



Manual

One Kit | Unlimited
Possibilities

www.Cretile.com

1. CRETILE	6
2. PROJECTS	22
Project-1: Automatic drawer light	22
Project-2: Sunlight based alarm	23
Project-3: Burglar alarm using motion sensor	24
Project-4: Open door alarm	25
Project-5: Automatic night lamp	27
Project-6: Automatic light [using light sensor and motion sensor]	28
Project-7: Sound operated switch	30
Project-8: Vibration alarm	31
Project-9: Wireless remote bell	32
Project-10: Wireless remote controlled vehicle	34
3. CRETILE PROLOGIC - INTRODUCTION	37
4. PROLOGIC - SOFTWARE INSTALLATION GUIDE	38
Installing Arduino software for Cretile	38
Installing USB drivers for Cretile ProLogic	40
5. CRETILE™ PROLOGIC - QUICK START GUIDE	42
Setting up software for programming	42
Introduction to ArduBlock	46
Developing first program for Cretile ProLogic	49
Cretiles Needed	49
Writing ArduBlock Blinking Light Program	50
Executing Blinking Light Program on ProLogic	56
6. PROLOGIC IN DETAIL	57
Cretile ProLogic Capabilities	57
Inputs and Outputs	57
Motor Control	57
Wireless Communication	57
ProLogic Development Environment	57
ProLogic & Arduino Programming using ArduBlock	58

ProLogic program structure (Setup & Loop)	58
Control structure	59
a. If then, if then else	59
b. Loops	60
c. While Loops	60
d. Repeat Loops	61
Variables	62
Time functions	63
Details of ProLogic blocks in ArduBlock	64
Working with ProLogic local inputs and outputs	64
a. READY ProLogic without remote	64
b. WRITE ProLogic local output	64
c. READ ProLogic local input	64
Examples: Local inputs and outputs	65
1. Blink	65
2. Setting speed of blinking using Dimmer and ProLogic	66
ProLogic remote inputs and outputs	67
a. Ready ProLogic remote IN	68
b. Ready ProLogic remote OUT	68
c. Ready ProLogic remote IN & OUT	68
d. WRITE ProLogic remote output	69
e. READ ProLogic remote IN	69
Examples: Remote inputs and outputs	69
1. Remote vehicle control using program	69
Using ProLogic motors	73
a. SET Motor Speed	73
b. SET Motor Direction	73
7. USING CRETILE WITH ANDROID	74
Introduction	74
Setting up App Inventor	74
Installing offline version of App Inventor	74
First time login to App Inventor	75
Creating project in App Inventor	77
Using Android App Inventor extension	78
Setting up your Android phone	81
Developing Cretile Android App in App Inventor	82
Cretille App Inventor blocks	82
a. ProLogic.Start	82
b. ProLogic.ConnectionStatusChanged	82
c. ProLogic.InputValuesReceived	83
d. ProLogic.READ_IN	83
e. ProLogic.READ_remoteIN	83
f. ProLogic.WRITE_OUT	84
g. ProLogic.WRITE_remoteOUT	84
h. ProLogic.SET_motorSpeed	84
i. ProLogic.SET_motorDirection	85
Developing android application	86
Installing application on android phone	92

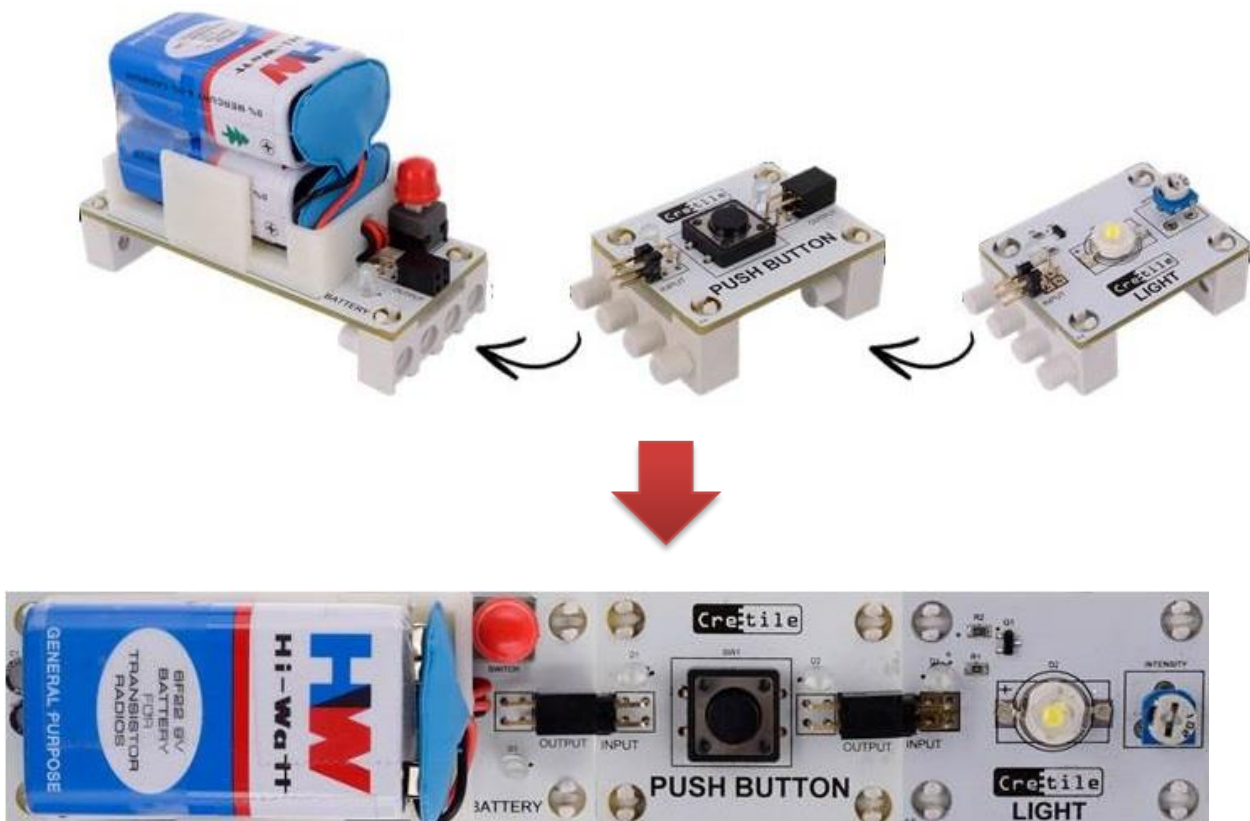
Developing ProLogic program to work with the Android application -----	95
Blocks for Mobile/PC communication in ArduBlock-----	95
Developing ProLogic Program-----	97
Connecting Android phone with ProLogic using USB-OTG-----	98
Start application created in section [Developing android application] in android phone and connect the phone with ProLogic which is programmed in section [Developing ProLogic Program]-----	98

1. Cretile

Cretile is a kit of modular electronics, electro-mechanical blocks which can be connected with each other like pieces of puzzle. There is no breadboard or soldering required for making circuits using Cretile.

Since there is no loose wire connection or breadboard is required to make connection, chances of loose connections of wires are removed. Also two modules will connect to each other only in correct manner and there is no possibility of wrong connections. Each module can connect to other module using an electrical connector so there is no need of soldering also to connect two modules.

Following image shows three modules before connection, their connection procedure and final connected modules.

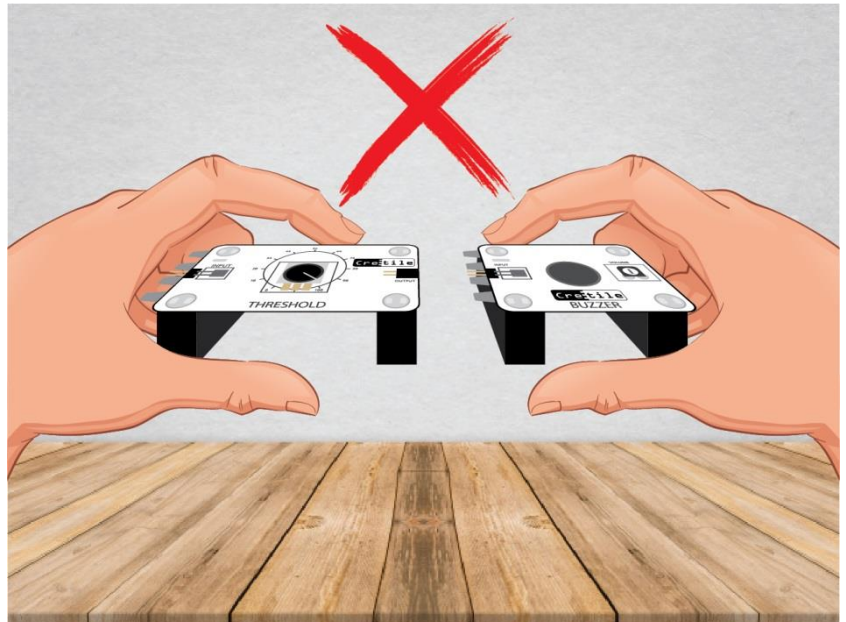


Dos and Don'ts while using Cretile

- 1 While connecting two Cretiles, place the Cretiles on a flat surface such as table and connect by sliding them against each other as shown here.



- 2 Do not try to connect Cretiles without any support of surface. This may damage the pins connectors of Cretile. Always place on a flat surface while connecting.

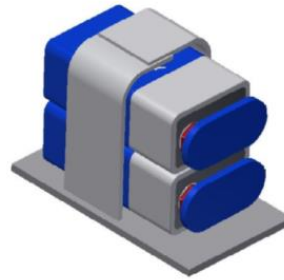


- 3 While disconnecting two Cretiles do not pull Cretiles forcefully from each other. Firmly hold one of the Cretiles and softly pull second Cretile out while moving it little up and down.
- 4 Always connect the Cretiles in switched OFF position.
- 5 Do not touch Cretile circuit part when in switched ON position. Do not bring in contact with any liquid or metal.

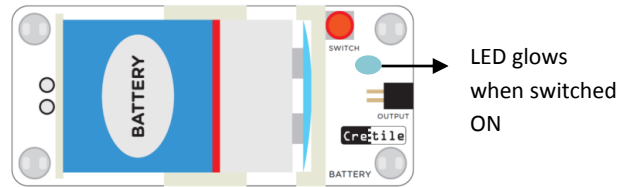
BATTERY

Cretile Battery gives power from battery to other Cretiles connected to it. You need to connect two 9V batteries to Cretile Battery.

- a) Connect the battery connectors to battery terminals as shown below.
- b) Place these batteries in battery holder and fasten as shown below.

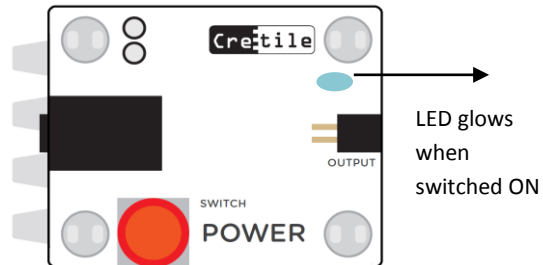


- c) Operate the button to switch it ON or OFF.



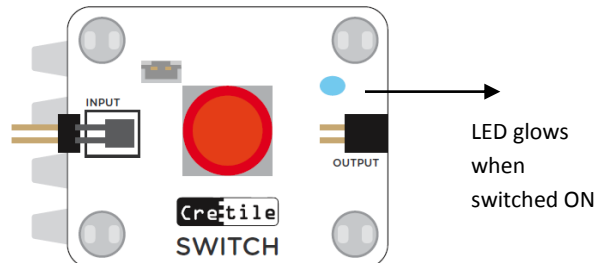
POWER

You can choose to use Power instead of Battery to provide power to assembly of Cretile connected to it. You will need 9V 2A adapter with compatible DC PIN to be connected at the power jack of Cretile Power.



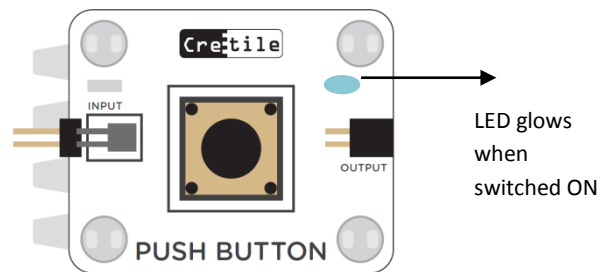
SWITCH

Cretile Switch allows you control Cretiles connected to it. It has one input and one output. The Cretiles connected on the output can be switched ON or OFF with the switch.



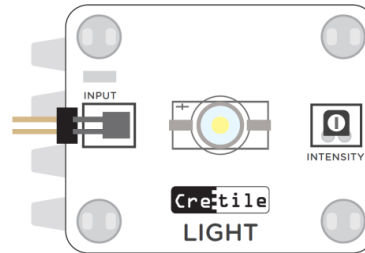
PUSH BUTTON

Cretile Push-button is similar to one used in door bell. It has one input and one output. It is by default OFF. Cretile Push Button remains ON for the duration the push button is pressed. Once the push button is released, it switches OFF.



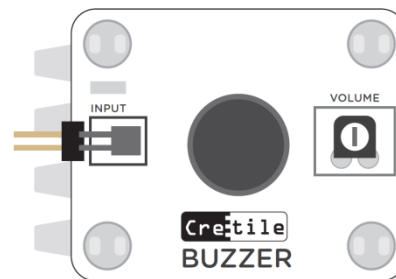
LIGHT

Cretile Light provides light for your creations. It has one input to switch the light ON or OFF.



BUZZER

Cretile Buzzer provides sound for your creations. It has one input to switch the buzzer ON or OFF.



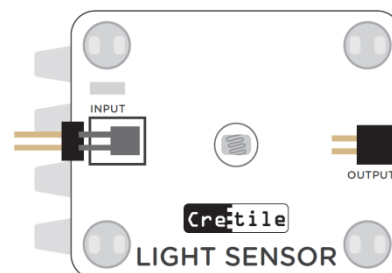
WIRE

Cretile Wire is used when you need to connect two Cretiles kept at a distance.



LIGHT SENSOR

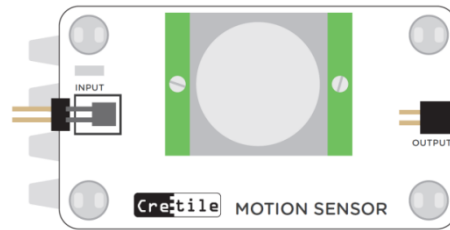
Cretile Light Sensor generates output based on amount of light shining on it. It has one input and one output.



Make & learn: Connect Battery + Light Sensor + Buzzer in sequence & power-on the assembly. Use a flash-light/torch and focus light on sensor module. You will observe that the Cretile Buzzer will sound as loudly as the amount of light shining on the Cretile Light Sensor.

MOTION SENSOR

Cretile Motion Sensor detects the motion around it. It has one input and one output. Output is turned ON when motion around it is detected. It turns OFF when motion stops.

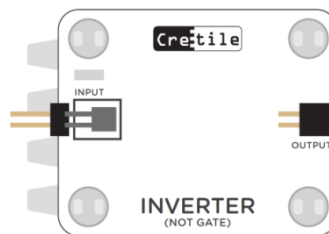


Note: After connecting to power, Motion Sensor needs about one minute time to start functioning correctly. Please wait for one minute after turning ON power. Also note that motion sensor can detect motion of human being or other life forms. It cannot detect motion of objects.

Also note that there it could take about 5 seconds for the motion sensor to detect new motion. So, expect a delay of about 5 seconds from motion to sensor switching ON.

INVERTER(NOTGATE)

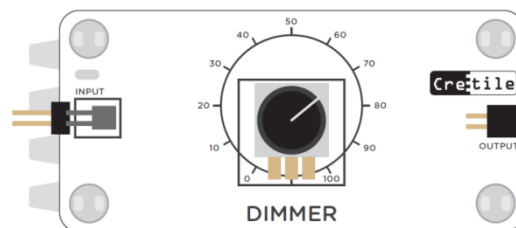
Cretile Inverter performs inverting operation. It has one input and one output. If input is ON, its output turns OFF. Likewise, when input is OFF, its output turns ON.



Make & learn: Connect Battery + Switch + Inverter + Light in sequence & power-on the assembly. When you operate Cretile Switch to ON position the Light switches OFF. Similarly, when you operate Cretile Switch to OFF position the Light switches ON.

DIMMER

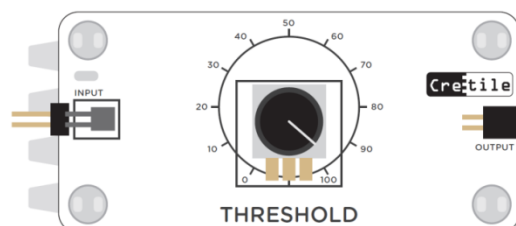
Cretile Dimmer is similar to fan speed regulators. It can change the speed of Cretile Motor or the intensity of the Cretile Light module etc.



Make & learn: Connect Power + Dimmer + Light in sequence & power-on the assembly. You can vary the intensity of the Light by operating the dimmer knob. Similarly, speed of Cretile Motor, sound of Cretile Buzzer etc. can also be controlled using Dimmer.

THRESHOLD

Cretile Threshold is useful in projects where you want action to happen only if input is more than a certain value. If input is less than dial position value, the output remains OFF. When



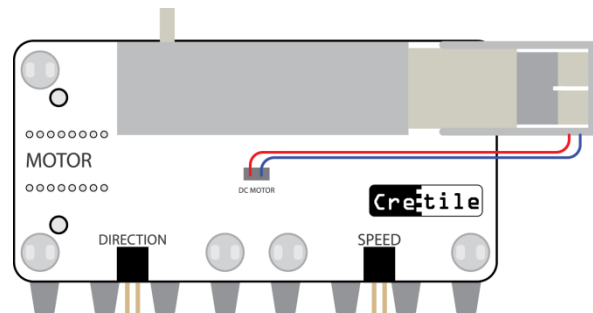
input is more than dial position value, output becomes ON.

Make & learn: Let's build a morning alarm that will ring only when the room brightness increases beyond a certain level. Here you will use Cretile Threshold to control the level of brightness beyond which the Buzzer should switch ON.

Connect Battery + Light Sensor + Threshold + Buzzer in sequence & power-on the assembly. Keep the Cretile Threshold module at say 20% position. Keep this assembly at a place where morning sunlight can fall on the Cretile Light Sensor. In the morning, when sunlight brightness increases beyond 20%, the buzzer will start ringing. If you increase Threshold to 30% then it will take more light for the alarm to turn ON.

MOTOR

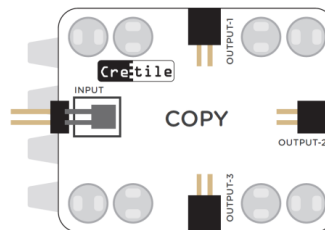
Cretille Motor provides motion to your creations. Cretille Motor has two inputs: one to control speed control & other to control direction. The Cretille Motor comes with a wheel and a pulley.



Make & learn: Connect Battery + Dimmer at 'Speed' input. Connect Cretille Switch at 'Direction' input and power-on the assembly. With Dimmer you will be able to change Motor speed & with Switch you will be able to change Motor direction.

COPY

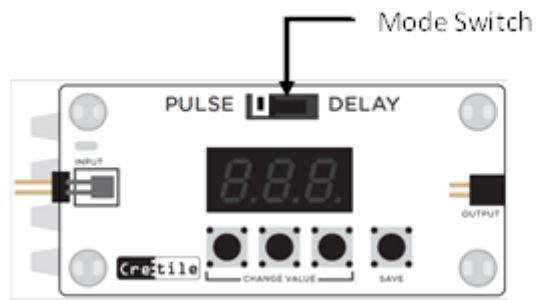
Cretille Copy is useful when you want to control two or more Cretilles with only one input. Whatever is supplied at input of Copy gets replicated at its three outputs.



Make & learn: Let's say you need to change both light intensity and motor speed with one dimmer. Connect Battery + Dimmer + Copy in sequence. Connect Motor speed input at one of the outputs of Copy. Connect Light at other output of Copy. Now, power on the assembly. As you operate the dimmer, light intensity and motor speed changes simultaneously.

PULSE/DELAY

Cretile Pulse/Delay can perform two functions: a Time Pulse or Time Delay. The functionality can be decided by using mode switch. It has one input and one output.

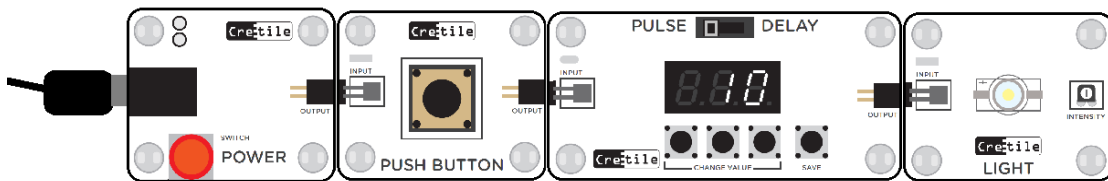


PULSE

Cretile pulse is used in situations where you want the action to happen only for a certain amount of time.

Make & learn: Let's make a project that keeps the Cretile Light switched ON for 10 Seconds only. Connect Power + Push Button+ Pulse/Delay + Light in sequence. Put the Cretile Pulse/Delay in Pulse mode. Set the Pulse value to 10 seconds with 'CHANGE VALUE' buttons. Use 'SAVE' switch to store the Pulse value for future use. Now operate Cretile Push Button once. You will observe that the Cretile Pulse starts counting down from 10 to zero. During this period the Cretile Light remains ON. Once the countdown reaches zero, the Cretile Light goes OFF.

Even though the Push Button is switched ON only for a moment, the switching ON action is stretched to the 10 seconds, as if you had pressed the push button for 10 seconds.

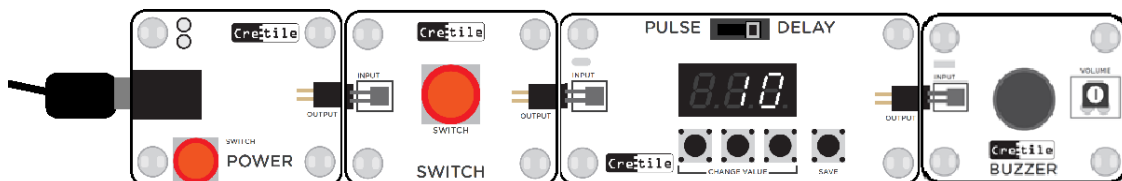


DELAY

Cretile Delay is used in situations where you want the action to happen after a certain amount of time.

Make & learn: Let's make a count-down alarm. Use Power + ON/OFF Switch + Pulse/Delay + Buzzer in sequence. Initially keep the Cretile Switch in OFF position. Set the Cretile Pulse/Delay to delay mode. Set the Delay value to 10 seconds with 'CHANGE VALUE' buttons. Use 'SAVE' switch to store the Delay value for future use. Now operate Switch to ON position. You will observe that the Cretile Delay starts counting down from 10 to zero. During this 10 second period the Cretile Buzzer remains OFF. Once the countdown reaches to zero, the Buzzer starts to ring.

You have delayed the action of switching ON the Buzzer by 10 seconds after operating the switch.



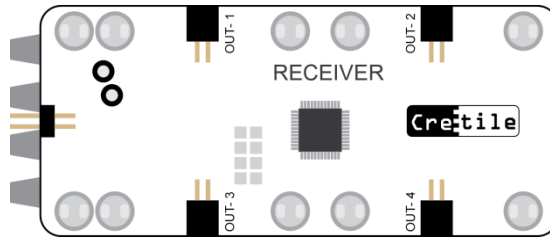
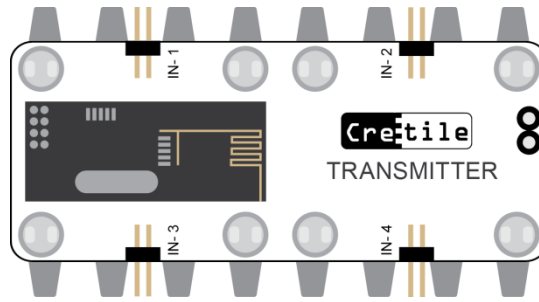
TRANSMITTER & RECEIVER

Cretile Transmitter and Cretile Receiver talk to each other wirelessly. It gives your projects ability to remotely control and take action.

Cretile Transmitter has four inputs. It sends these four input values wirelessly to Cretile Receiver.

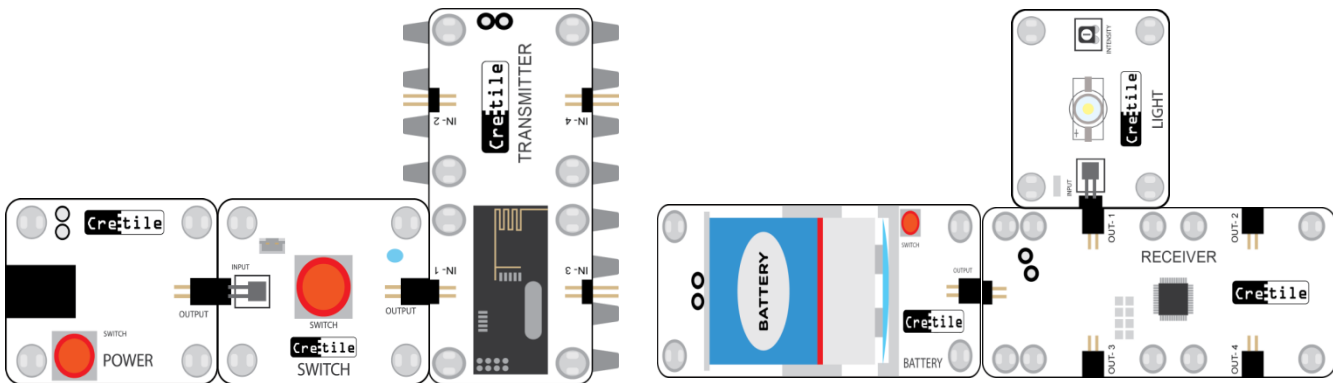
Cretile Receiver is always listening to messages from Cretile Transmitter. Four outputs of Receiver reflect what is received by it.

Cretile Receiver and Transmitter need separate power supply. You can use either Cretile Power or Cretile Battery for power supply.



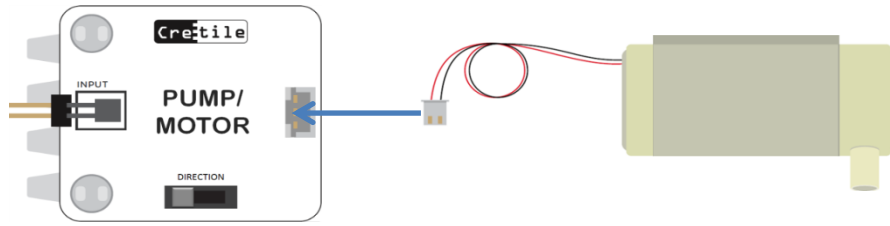
Make & learn: Connect Cretile Switch to one of its inputs say Input-1 of Cretile Transmitter. Connect Cretile Light to corresponding output, in this case, Output-1 of Cretile Receiver. Now operate the switch on the Cretile ON/OFF switch to ON position. It will switch ON the Cretile Light connected at Receiver.

You can connect any four inputs at Transmitter to control corresponding four outputs connected at Receiver.

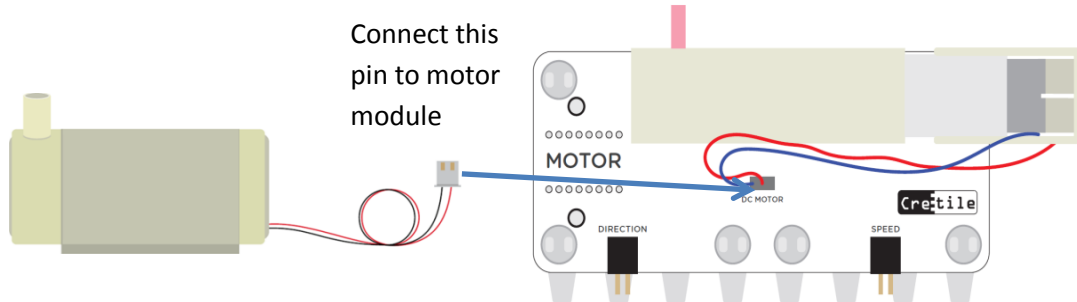


SUBMERSIBLE WATER PUMP

Submersible water pump can be used for pumping the water. The pump has to be immersed completely inside water. This pump has to be used with Cretile Pump/Motor as shown below.

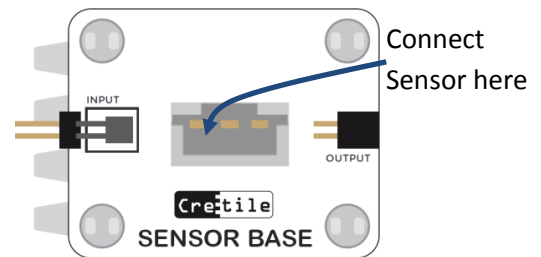


Alternatively, you can drive the Cretile Pump using Cretile Motor module as well.



SENSOR BASE

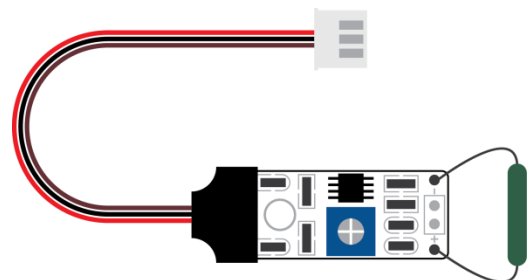
Cretile Sensor Base allows you to make use of various sensors in your project. You can connect variety of sensors (one at a time) to Cretile Sensor Base. Sensors such as Obstacle Sensor, Line Following Sensor, HAL Magnetic Sensor, Sound sensor etc. can be connected to Sensor Base.



MAGNETIC SWITCH (REED SWITCH)

To use Cretile Magnetic Switch, connect it to the Cretile Sensor Base. By default Magnetic Switch remains ON.

It switches OFF when a magnet is brought close to it within a few millimeters. Note that it will work with only one of the sides of a magnet. Try both sides of the magnet to check which side works. Use it with Inverter to keep OFF by default and switch ON when magnet is brought closer to it.



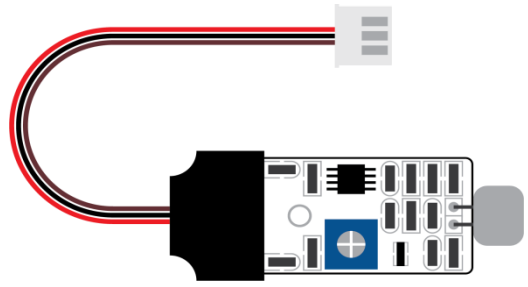
SOUND SENSOR

To use Cretile Sound Sensor, connect it to the Cretile Sensor Base.

Cretille Sound Sensor detects sound around it. The sensor by default remains switched ON. It switches OFF when sufficiently loud sound is detected around it.

The sensor remains OFF only momentarily for duration of sound detected. E.g. when you clap around the Sound Sensor it remains ON for a short duration of the clap.

You can decide the sensitivity of the Sound Sensor by operating the potentiometer placed on it. Use a small screw-driver to operate this potentiometer. Use it with Inverter to keep OFF by default and switch ON when sound is detected.



Make & learn: Connect Cretille Sound Sensor to Cretille Sensor Base. Now, connect Battery + Sensor Base (with Sound Sensor mounted on it) + Inverter+ Pulse-Delay + Light in sequence. Put the Cretille Pulse/Delay in Pulse mode. Set the Pulse value to say 600 seconds with 'CHANGE VALUE' buttons. Use 'SAVE' switch to store the Pulse value for future use. Now clap near the Sound Sensor. You will observe Sound Sensor detects the clap sound and gives ON Signal to Cretille Pulse. The Cretille Pulse starts counting down from 600 to zero. During this period the Cretille Light remains ON. Once the countdown reaches zero, the Cretille Light goes OFF.

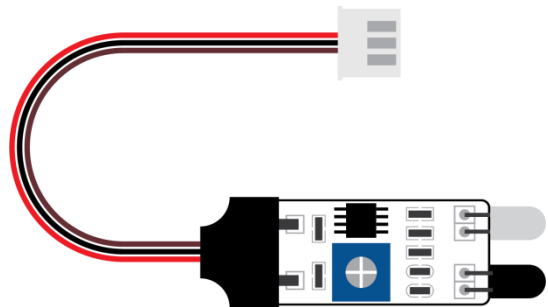
OBSTACLE SENSOR

To use Cretille Obstacle Sensor, connect it to the Cretille Sensor Base.

The sensor by default remains switched ON. It switches OFF when it comes sufficiently close to an object. Sensor can detect objects that are within few millimeters to a couple of centimeters from it. This detection distance can be changed by turning potentiometer (dimmer) mounted on sensor.

Please note that the obstacle sensor does not work with black surfaces. When brought close to an object with black surface, it may not switch OFF. It may switch OFF when it is very close to a black object.

Use it with Inverter to keep OFF by default and switch ON when it approaches an obstacle.



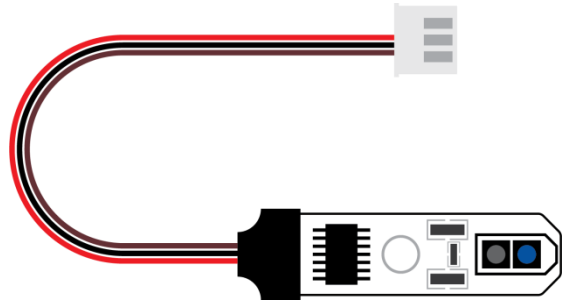
LINE-FOLLOWER SENSOR

To use Cretile Line-Follower Sensor, connect it to the Cretile Sensor Base.

Cretile Line Follower sensor is specially designed to detect black surface against any other color surface.

The sensor remains ON when there is nothing in front of it. It switches OFF when black surface/line is detected within few millimeters.

Use it with Inverter to keep OFF by default and switch ON when it detects black line. Note that the sensor should always be kept parallel to a surface within few millimeters. Sensor should not completely touch or too far away from a surface.



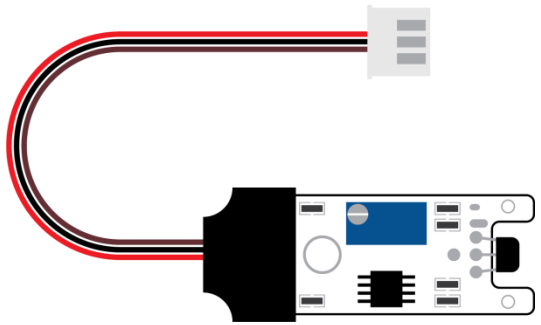
MAGNETIC SWITCH (HALL EFFECT)

To use Cretile HALL Effect Sensor, connect it to the Cretile Sensor Base.

Cretile Magnetic Hall effect Switch generates output based on how close a magnet is placed close to sensor.

It will work with only one of the sides of a magnet. So, try both sides of the magnet to check which side works. The closer the magnet, higher the output that sensor generates. As you take the magnet away from sensor, it generates lesser output.

Make & learn: Connect Cretile Hall Sensor to Cretile Sensor Base. Now, connect Battery + Sensor Base (with Hall Sensor mounted on it) + Light in sequence. Now bring a strong magnet close to the sensor. You will observe that the Light glows with high intensity if magnet is close to sensor. Similarly the Light glows with lesser intensity if magnet is taken away from the sensor.

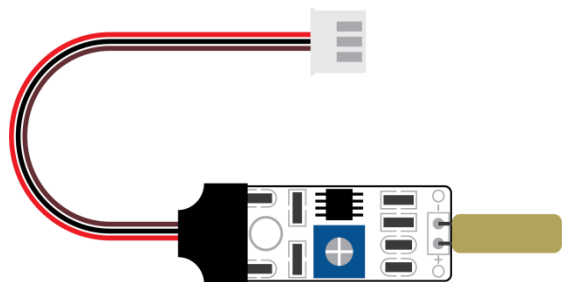


TILT SENSOR

To use Cretile Tilt Sensor, connect it to the Cretile Sensor Base.

Cretile Tilt Sensor by default remains ON when placed on a flat surface. It is switched OFF when the sensor or surface on which it is mounted is tilted.

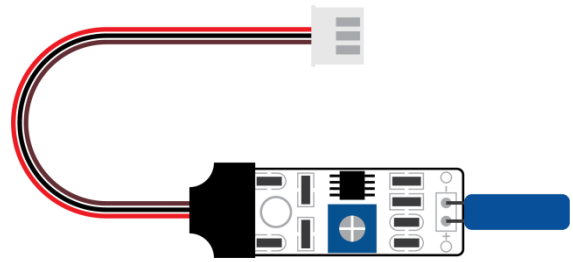
Use it with Inverter to keep the sensor OFF by default and ON when it detects a tilt.



VIBRATION SENSOR

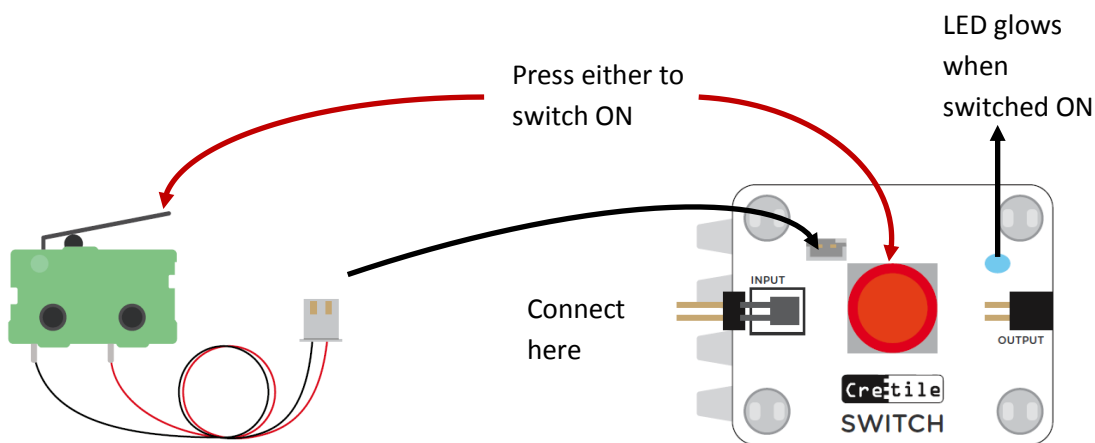
To use Cretile Vibration Sensor, connect it to the Cretile Sensor Base.

By default, Cretile Vibration Sensor remains OFF. It is switched ON when vibration is detected on the surface on which it is placed. E.g. when surface on which the sensor is placed is knocked with hand.



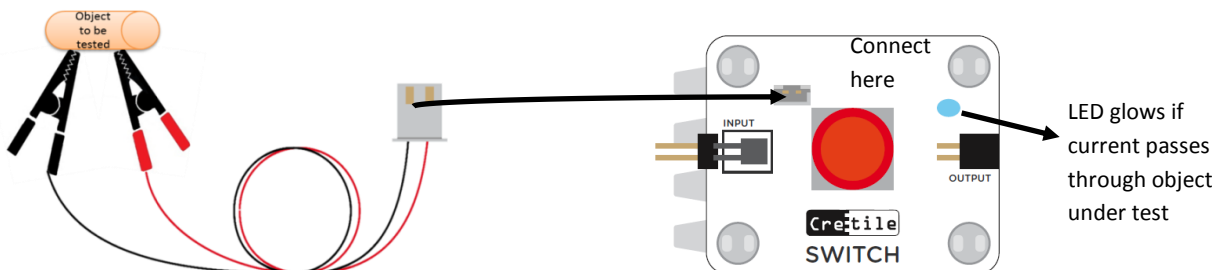
LIMIT SWITCH

Cretil Switch allows you to switch ON or OFF the other Cretile connected to it. Cretil Switch can also be extended to be used with operated using a Limit Switch as shown below. This way the Cretils connected to Switch can now be switched ON or OFF either by operating the button on the Switch or by operating the Limit switch.



CONNECTIVITY

You can use Cretil Connectivity to test electrical conductivity of an object. Connect the Cretil Connectivity to Cretil Switch as below. Now connect pins on two sides of the object to be tested. If the object is good conductor of electricity then Cretil Switch turns ON.

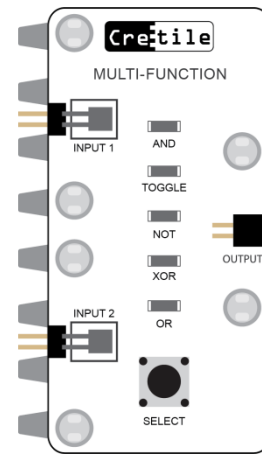


Make & learn: Connect Battery + Switch with Connectivity+ Buzzer in sequence & power-on the assembly. Connect the pins to either side of an iron metal piece. You will observe that the Cretil Buzzer will start ringing as metal is conductor of electricity. Connect the pins to either side of a plastic piece. In this case the Buzzer does not ring as plastic is a bad conductor of electricity.

MULTI-FUNCTION

Cretile Multi-Function provides functions of Logic Gates and Toggle Function. Logic Gate Functions such as (1) AND (2) NOT (3) XOR & (4) OR is available. With Toggle function you can change the output from ON to OFF or from OFF to ON with a switch connected at Input 1.

SELECT switch provided on the module can be used to decide the function. Below table shows which inputs are used by different function



Function	Input - 1	Input -2
AND	Yes	Yes
TOGGLE	Yes	No
NOT	Yes	No
XOR	Yes	Yes
OR	Yes	Yes

Logic Gates functionality

AND			NOT			XOR			OR		
IN1	IN2	OUT	IN1	IN2	OUT	IN1	IN2	OUT	IN1	IN2	OUT
ON	ON	ON	ON	-	OFF	ON	ON	OFF	ON	ON	ON
ON	OFF	OFF	OFF	-	ON	ON	OFF	ON	ON	OFF	ON
OFF	ON	OFF				OFF	ON	ON	OFF	ON	ON
OFF	OFF	OFF				OFF	OFF	OFF	OFF	OFF	OFF

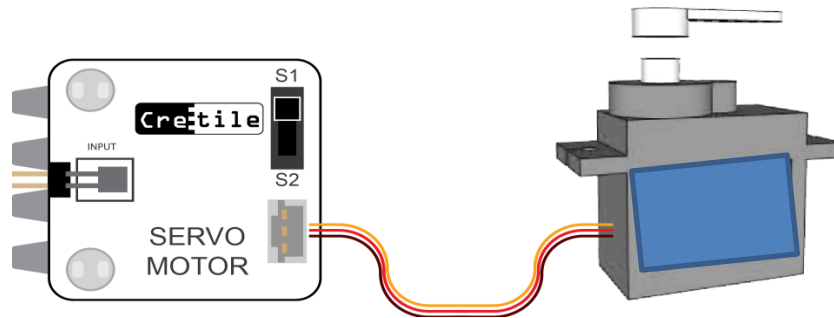
Toggle Function:

Toggle reverses the current output. So, if the current output is ON, after giving input-1 as ON, the output will be turned OFF and vice-a-versa.

Make & learn: Connect Cretile Push Switch to Input-1 and Cretile Light to Output-1. Initially the Light will remain off. Once you press the Push Switch the Light will switch ON. If you press the Push Switch, the output will switch OFF. So, output toggles between ON and OFF.

SERVO MOTOR

Cretile Servo Motor can be used in projects where you need to motor arm to reach to a specific location e.g. in project where you need to a waving flag you can mount a flag on servo motor and control the motor movement.

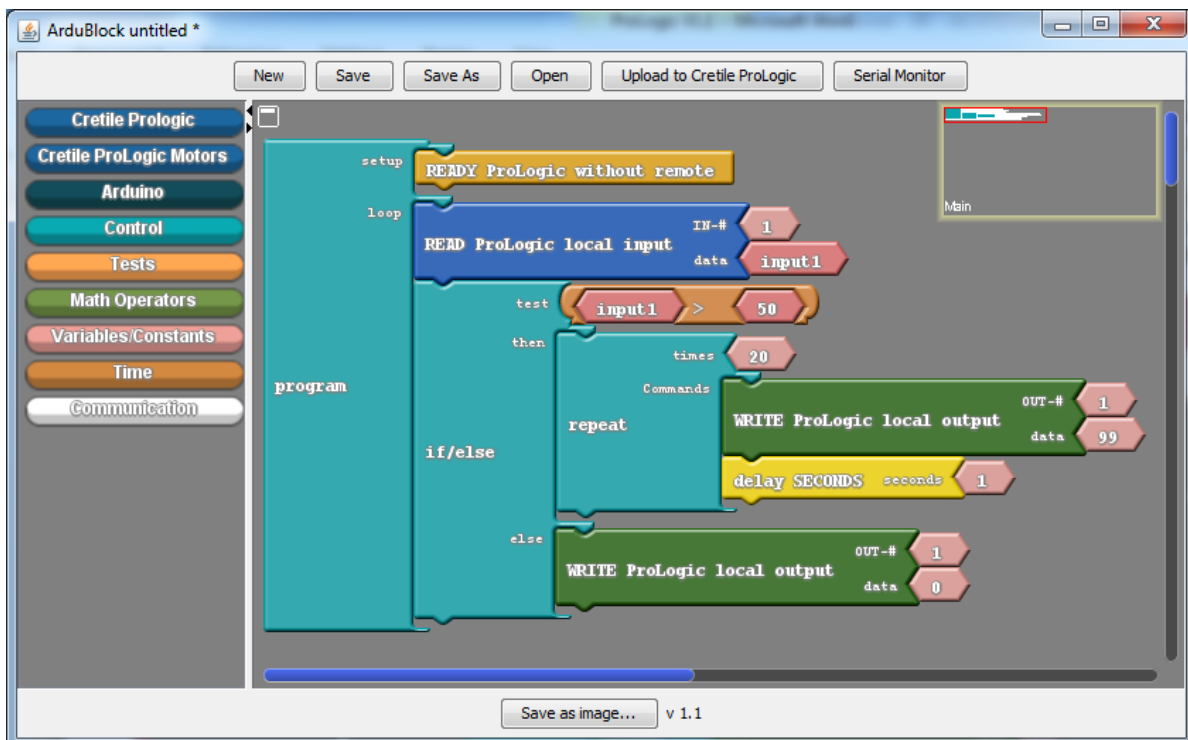
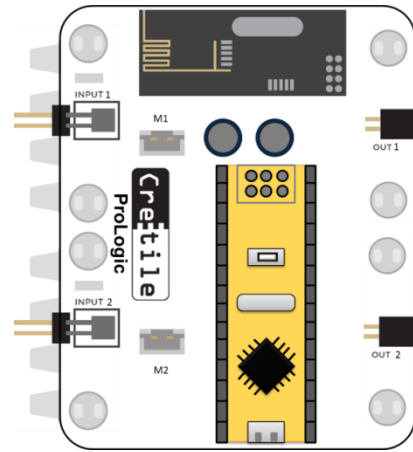


PROLOGIC

Cretile ProLogic is a programmable module. It can be programmed with the help of ArduBlock* or by writing C/C++ code.

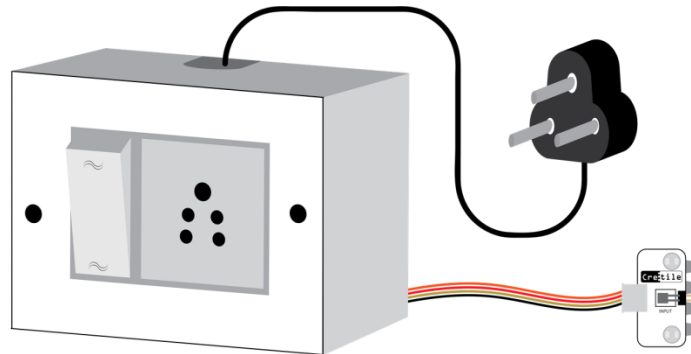
ArduBlock provides a simple drag and drop mechanism to build a program Logic.

The ArduBlock environment looks like below



SMART-SWITCH

Cretile SmartSwitch allows you to control 240V/5A appliance to be controlled with Cretile inputs. Cretile SmartSwitch looks like a regular wired extension box. It has one Cretile input. You can control an appliance connected to SmartSwitch with Cretile logic connected at this Cretile input.



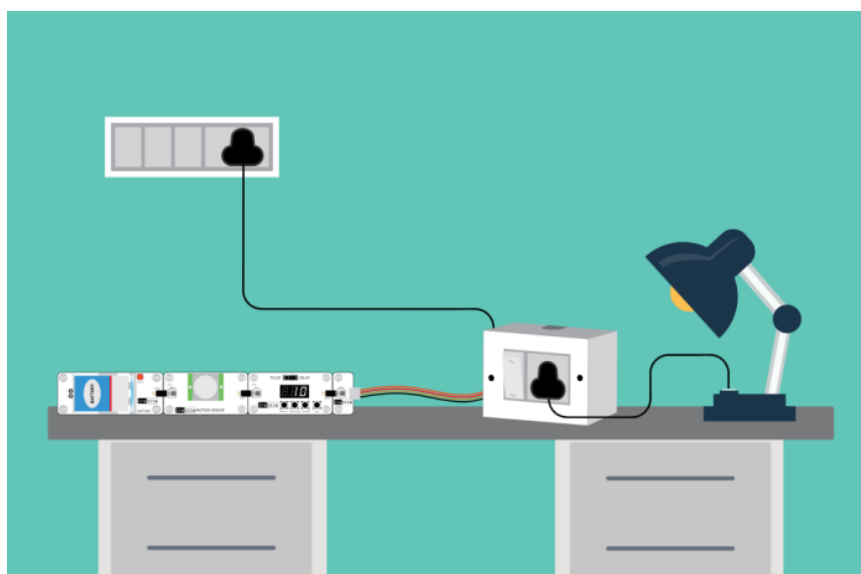
Make & learn: Let's control a table lamp with Motion detection. Make the connections as below. Connect the table lamp at SmartSwitch socket.

Connect Battery + Motion Sensor + Pulse Module in sequence at Cretile input. Connect the power input to 240V wall power and switch ON the power supply. You would observe that the table lamp does not switch ON.

Now switch the Cretile Battery Power ON. Now Motion Sensor starts to detect the motion around it. When it detects the motion it switches ON the table lamp. When no motion is detected, it will switch OFF the table lamp.

Note: Motion Sensor takes about a minute to start functioning correctly. After powering ON, wait for about a minute before you expect your Motion Sensor based projects to function correctly.

With Cretile Pulse, the table lamp will remain switched ON for the set Pulse duration every time the motion is detected



2. Projects

Project-1: Automatic drawer light

Need some extra light to quickly find stuff in your drawer? Let's make a drawer light that switches ON automatically moment you open it.

DISCUSSION

Do you recall a shop/mall door opening automatically when you came close to door? Have you been to any place where the lights switched-on automatically when you entered room?

Find out the reason for these automatic actions: Your presence? Or is it your weight? Or is it sound?

If a sensor can detect presence of person then we can make automatic drawer light using the sensor. Can you think how?

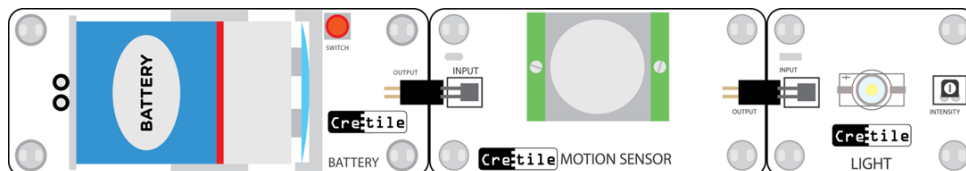
ACTIVITY

Let's make automatic drawer light now using Cretiles.

CRETILES NEEDED

- Power or Battery
- Switch
- Motion Sensor
- Light

Connect the Cretiles as shown below & keep this assembly in a drawer and wait for about two minutes.



When you open the drawer, the Cretile Motion Sensor detects your motion and switches the Cretile Light ON. The Light will switch OFF automatically once you close the drawer. It might take a few seconds to switch OFF the Light.

Note that after switching ON power, Motion Sensor needs about one minute time to start functioning correctly. Please wait for one minute after turning ON power.

Also note that there it could take about 5 seconds for the motion sensor to detect new motion. So, expect a delay of about 5 seconds from motion to sensor switching ON.

Project-2: Sunlight based alarm

Goal of this project is make an alarm based on sunlight. That means alarm will start in morning when there is sufficient sunlight.

DISCUSSION

How can we detect sunlight? Which sensor can we use for this?

How can we set brightness of light at which alarm should start?

ACTIVITY

Let's make sunlight alarm.

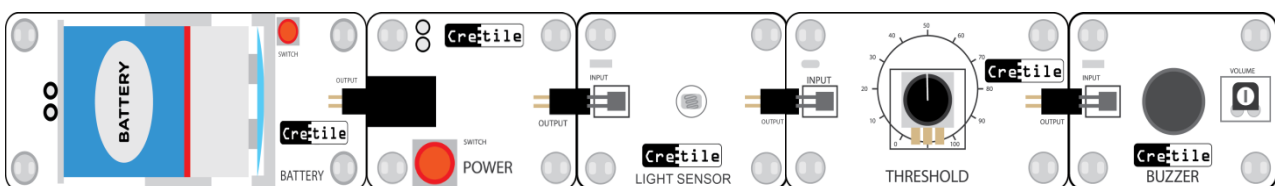
CRETILES NEEDED

- Power or Battery
- Switch
- Light Sensor
- Threshold
- Buzzer

Connect the Cretiles as shown blow. Keep this assembly at a place where sunlight can shine on the light sensor. Keep the Threshold module at say 70% position. In the morning, the Cretile Buzzer will start ringing when sufficient sunlight falls on the light sensor.

When the light increases beyond Threshold value, Threshold goes to ON position. This switches ON the Buzzer. The level of sunlight at which Cretile buzzer should turn ON can be decided by changing the value of Threshold module. If you want to get up earlier then reduce the Threshold value. If you want to get up later then increase the Threshold value.

The alarm can be switched off by operating Cretile Switch to OFF position.



Project-3: Burglar alarm using motion sensor

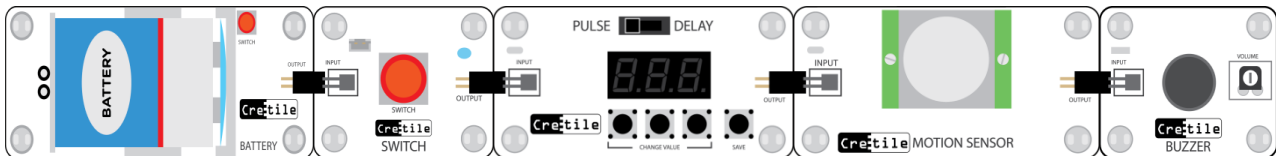
Goal of project is to sound alarm when any motion in room is detected after 1 minute of switching on the security alarm circuit.

DISCUSSION

How can we detect presence of person? What kind of sensor will be needed?

How can we delay functioning of burglar alarm?

ACTIVITY



- To power the circuit, we will need battery of power module.

To delay the security alarm circuit for 1 minute after switching on, we will need delay module.

- To detect any motion we will need motion sensor.
- To produce sound when motion is detected, we will need buzzer module.
- We need to connect these modules in given sequence above.
- When we power on the circuit ON/OFF switch should be in off position.
- Put pulse delay module in “delay” mode.
- Set the value to 60 seconds delay. Now when you press the switch to on position Delay for 60 seconds will start and after 60 seconds motion sensor will get activated.
- If there is any motion Buzzer will start to make sound since Buzzer is connected to output of motion sensor. Thus this project works as security alarm.

Project-4: Open door alarm

We will create an alarm which will start if door is left open for certain duration, for example 2 minutes.

DISCUSSION

Sometimes we forget to close a door and it can result in many unwanted situations like insects/mosquitos coming inside house. It can be security problem also as burglars can get access to house easily. If door has an alarm which will warn us about door left open then it can be very useful. There can be many ways such alarm can be made; can we make such alarm using a Dimmer?

ACTIVITY

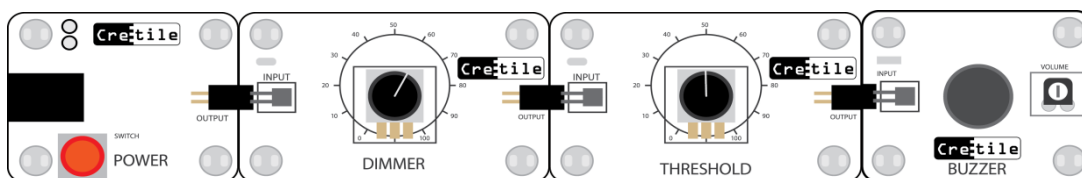
Let's make door open alarm using Dimmer as sensor.

We will need following modules for this

1. Battery module (power source)
2. Dimmer
3. Threshold
4. Pulse/Delay
5. Buzzer

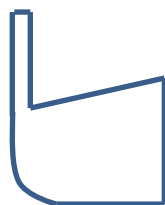
First we need to detect if door is open or closed. To detect if door is open, there are multiple ways. We will use dimmer to this. We can also use limit switch to do this but by using dimmer, we will learn principle of working of dimmer as sensor.

Connect the Cretile module as shown below

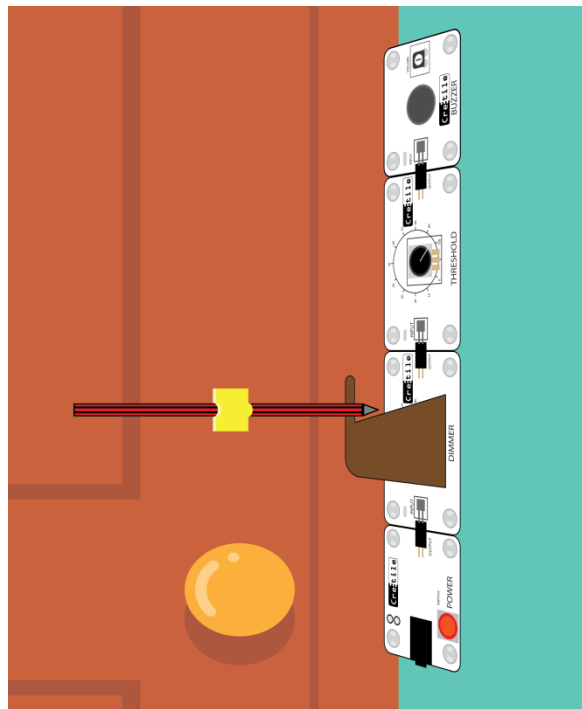


We need to attach a cardboard piece to Dimmer which will rotate dimmer when door is opened or closed.

Following image shows shape of the cardboard that we need to attach to dimmer.



Attach this cardboard shape to dimmer as shown in below image. And also attach complete assembly to door as shown in image below. This assembly can be attached to door which works using hinges and cannot be attached to sliding door.



Now whenever door opens it will push the cardboard and dimmer will rotate in clockwise direction and its output will increase. We will keep the threshold setting in Threshold module such that even slight increase in dimmer output will make Threshold on. So when door opens Threshold will switch on. This will start Delay timer which we have set to 120 seconds. If door remains open more than 120 seconds buzzer will switch on. If door closes before 120 seconds, pen on the door will push cardboard and dimmer will rotate anticlockwise and its output will reduce. Threshold will switch off and Delay timer will cancel and buzzer will not switch on.

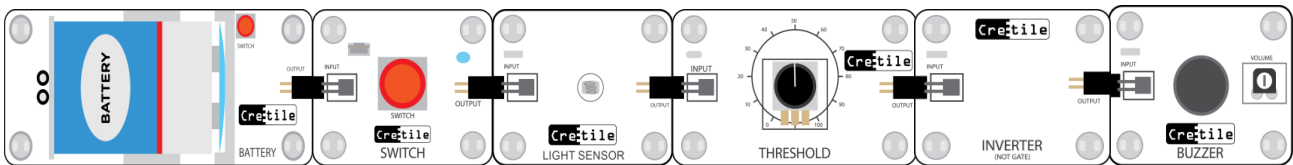
EXPLORE MORE

Let's modify the above circuit where dimmer and threshold is replaced by a limit switch. How can we make such door open alarm?

Project-5: Automatic night lamp

Connect the Cretiles as shown below. Keep the Threshold at say 30% position. At night when it gets dark, the Cretile light will switch ON automatically. The level of darkness when Cretile light should turn ON can be decided by changing the value of Cretile threshold. When the light level falls below Threshold value, Threshold goes to OFF position. Inverter inverts the OFF at its input to ON at its output. This switches ON the Light.

You can manually switch off the light by operating the Cretile Switch to OFF position.



Project-6: Automatic light [using light sensor and motion sensor]

Let's build a light which will switch ON automatically when light level is below certain threshold. The light will switch on only when someone is present in room.

DISCUSSION

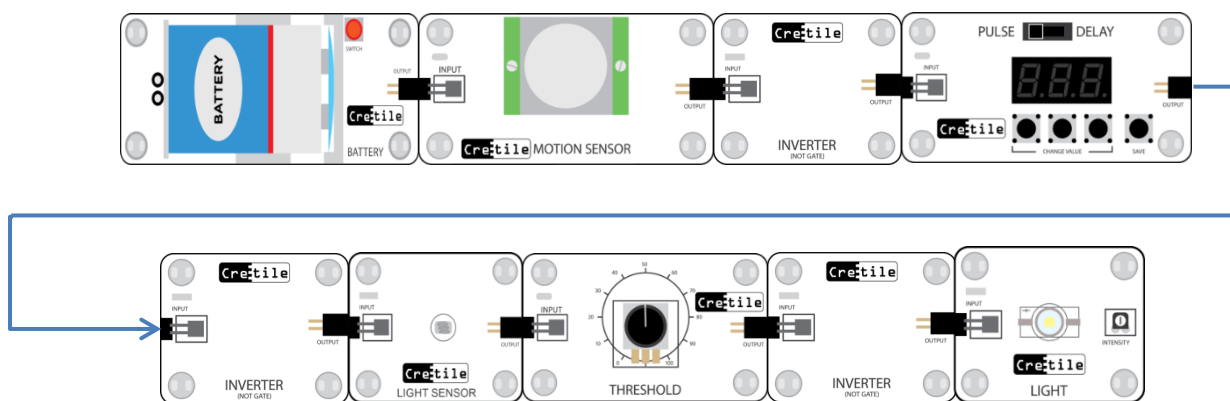
This type of automatic light is commonly used in homes and office conference rooms. By building this type of light we will understand component required building it, working principle and various settings required to suit ones need.

ACTIVITY

First we will identify all modules are required to build such light.

Just like any device or equipment we will need power source. We can use wall adapter and power module or battery module as our power source. Since we need to measure intensity of ambient light in room we will need light sensor module to measure it. We need feature to allow user to decide intensity of light at which this Light should switch on or off. This can be done by Threshold module. Finally to keep light switched on for certain duration after no motion is detected pulse/delay module will be required. This is required since we don't want to switch off light immediately after no motion is detected because person may be in room but sitting still. In that case we would like to wait for some time before switching off light.

Following image shows connection of these modules to make such light.



WORKING OF THE AUTOMATIC LIGHT

- Battery provides power to complete circuit
- Motion sensor can detect human motion, to detect "No motion" we need to connect inverter to motion sensor.
 - When motion sensor detects motion its output becomes on. When there is no motion its output switches off.

- We connect inverter to motion sensor. Now when there is motion, output of inverter will be off and when there is no motion output of inverter will be on.
- Thus we will get a “No Motion” sensor
- Delay will get triggered when there is no motion for some time. Delay will count down time till there is “no motion”. As soon as there is motion during this delay period it will stop counting and reset to original number it was set to. If there is no motion in this delay period Delay output will become on.
- We have connected inverter to output of delay. So when output of delay is on, output of this inverter is off. When output of delay is off, output of this inverter is on. Now due to no motion if Delay timer count downs to 0, its output will be on. Hence output of inverter will become off. As soon as there is motion detected again delay output will become off and inverter output will be on.
- Whenever input to light sensor is on, it can sense intensity of light. Its output will change according to intensity of light. If light sensor input is off, it will stop sensing light and its output will remain off. So when there is no motion for time period as set in Delay module, its input will be off and it will stop sensing light and output will become off. If there is motion, input of light sensor will be on and hence light sensor can sense light and its output will change according to intensity of light.
- Now using threshold we can decide the intensity of light at which Light should switch on.

Project-7: Sound operated switch

Goal of sound operated switch is switch ON or OFF light with clap or some other loud sound (for example whistle). Initially when circuit is powered on light should be off. When we clap if light is off it should switch on and if light is on it should switch off.

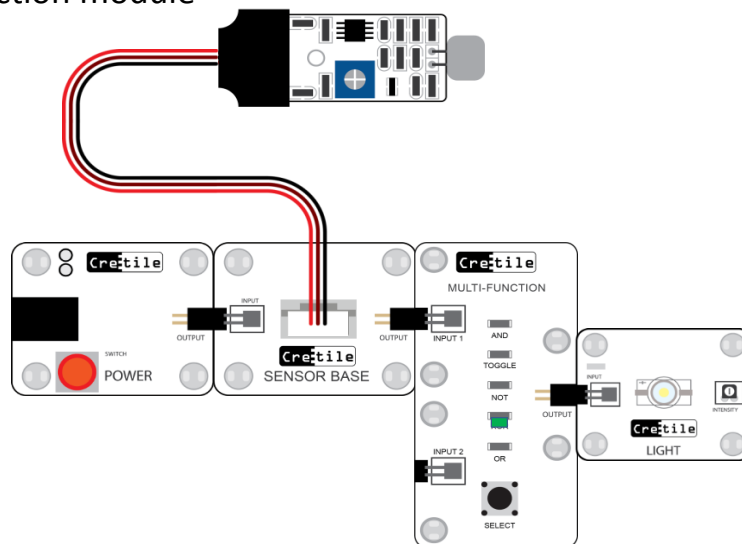
DISCUSSION

With remote operated appliances we need to always carry remote with us. Many times remote is misplaced and temporarily not available. In such situations we cannot switch on or off such appliance or lights. Will it be nice idea to have some switch which will be on or off with clap?

ACTIVITY

We need following modules for making this circuit

1. Power module
2. Sound sensor & Sensor Base
3. Multifunction module
4. Light



Power module provides power supply to circuit. Sound sensor detects sound and switches on for duration when sound is more than certain level as set by potentiometer on the sound sensor module. For example when we clap sound sensor will switch on and then switch off. We have connected Multifunction module to output of sound sensor. Use select switch on Multifunction module to select Toggle functionality. When output of sound sensor switches ON from OFF state, toggle module will change its output. Hence when light is off and we clap then light will switch on and vice versa.

Project-8: Vibration alarm

Goal of vibration alarm is to warn whenever there are vibrations at place where it is not expected.

DISCUSSION

Where can such vibration alarm be useful?

For example, generally a stable machine platform will not have vibration but when its mechanical parts start to malfunction there will be vibrations and if we want to warn for such situation then this circuit can do that.

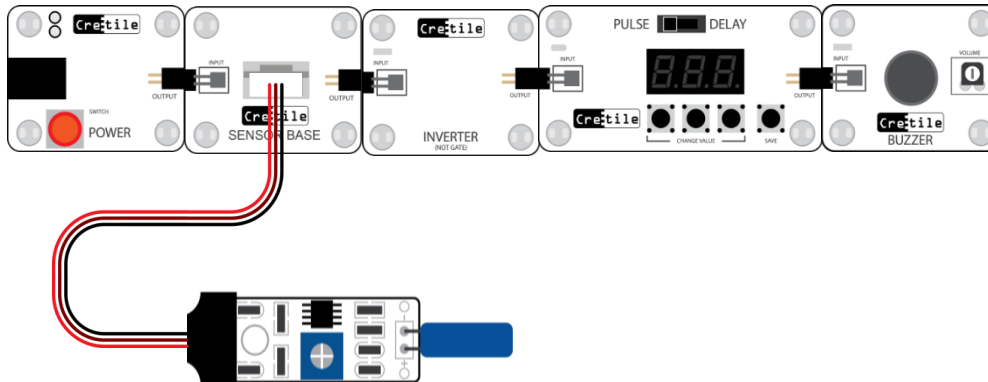
We can also use vibration alarm as security alarm. We can attach this to any artifact which we want to secure. Now whenever there is any movement of artifact, alarm will sound.

ACTIVITY

For Vibration alarm we will need following modules

1. Power module
2. Vibration sensor & Sensor Base
3. Inverter
4. Pulse/Delay
5. Buzzer

Connect the modules as shown below



Power module provides power supply to circuit. Vibration sensor detects the vibration and reports to the inverter, when the amount of vibration is more than the inverter provides signal to the pulse/delay module than the buzzer is connected to the output of the pulse/delay to turn on buzzer for specific time. Like this whenever the vibration is detected in the sensor buzzer will turn on automatically for certain time span. The sensitivity or calibration of vibration sensor can be adjusted in the preset which is in the vibration.

Project-9: Wireless remote bell

Generally every home has doorbell. We will learn how to make wireless doorbell.

DISCUSSION

We need not to do any wiring on walls for wireless remote bell to work. This project we can use in many places like schools, colleges, offices and home.

ACTIVITY

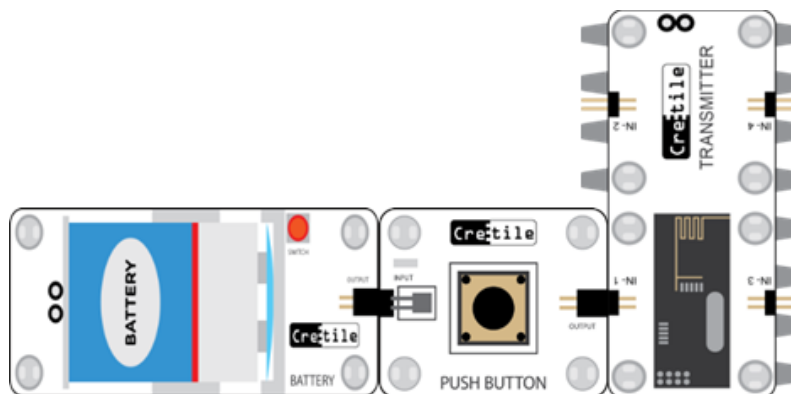
Following blocks are used in this project.

1. Battery
2. Button
3. Transmitter
4. Receiver
5. Buzzer

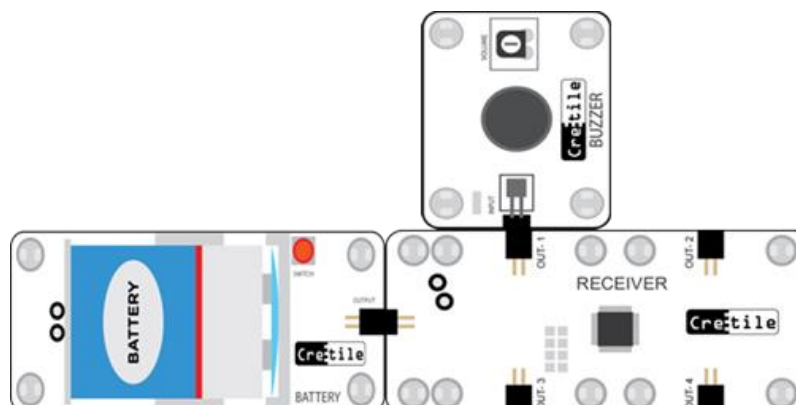
There are two assemblies in the project.

- 1) Transmitter
- 2) Receiver

To prepare transmitter connect Cretile power to ON/OFF switch then connect push button at last connect a transmitter at IN-1.



To prepare receiver, connect Cretile power to Receiver. Then connect Buzzer at Receiver's OUT-1 as shown below.



Power on batteries (power) connected to both Transmitter and Receiver assembly. When we press the push button connected to Transmitter, it will send the signal to the receiver and receiver OUT-1 will be on. Hence Buzzer connected to receiver will switch ON.

Project-10: Wireless remote controlled vehicle

DISCUSS

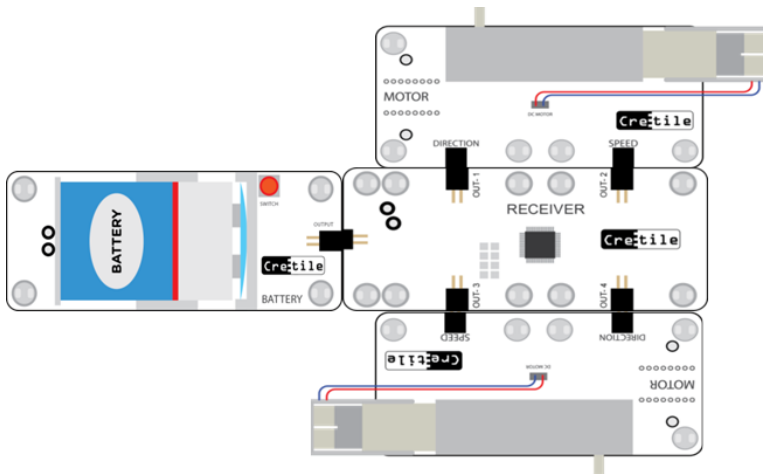
1. Robotics is emerging technology area. Robotics has applications in many different areas like defense, manufacturing, agriculture, rescue etc. Many robotics applications need moving vehicle platform. In this activity we will make such remote controlled vehicle.
2. How to prepare a remote controlled vehicle? What components will you need and why?
3. Where such remote controlled vehicle can be used?

CRETILES NEEDED

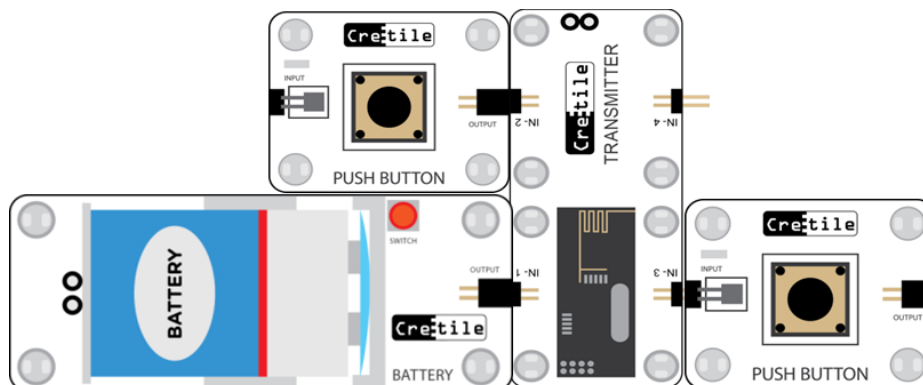
1. Power
2. Push Buttons - 2
3. Transmitter
4. Receiver
5. Motors

Connect the Cretile blocks as shown in both diagrams.

3. Below diagram shows vehicle side of Cretile assembly.



4. Below diagram shows remote control side of Cretile assembly.



- After the connections as shown above
- Use the two push buttons to control the robot
- If we press one button robot moves in one direction,
- If we press both buttons robot will rotate in one direction continuously

1) Think why robot is not moving in one direction like forward or reverse?

Two motors are rotating in opposite direction so robot is rotating fast at same place instead of moving forward.

2) What should be done for the robot to move forward?

As both the motors are rotating in opposite direction we need to change direction of one motor to make robot move

3) What is to be done to change direction of one motor?

We know that each motor is having two ports

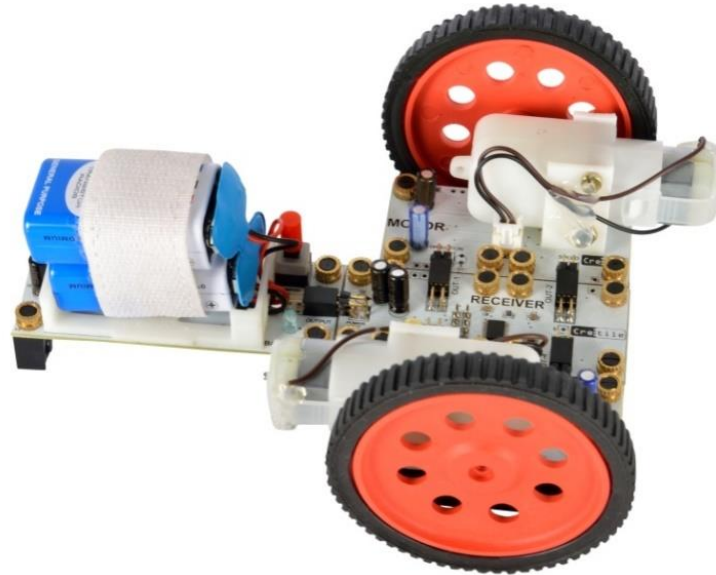
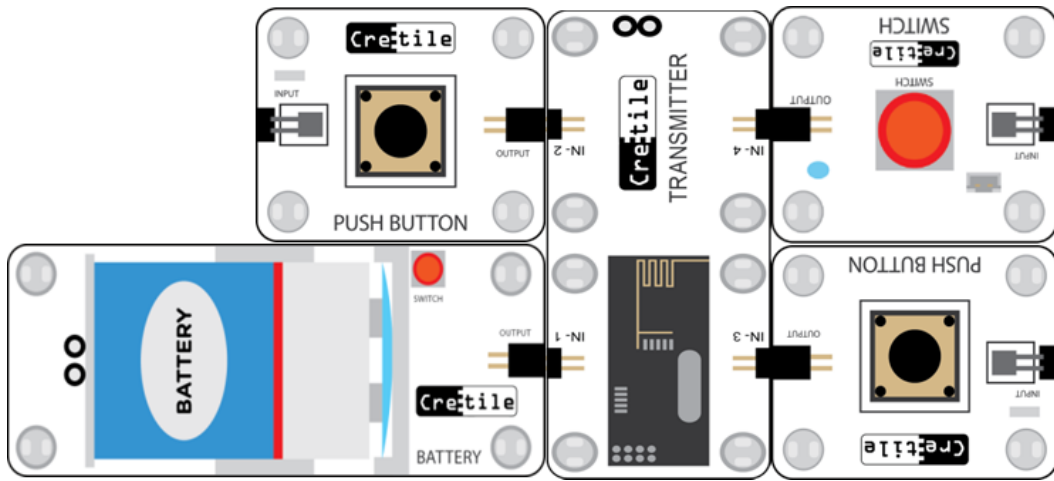
- 1) Speed
- 2) Direction

If direction port is not given with any input, by default it will be ON that makes the motor to rotate in one direction

Now in transmitter connect one Switch to port 1 or 4 and keep it in OFF position. For one of the motor, direction will change depending upon where you connect Switch

So we can observe that if we press two push buttons, both motors rotate in one direction. In this way we can prepare a robot which can move in forward direction.

Below diagram will show the setup of transmitter assembly to move robot in forward direction.



3. Cretile ProLogic – Introduction

ProLogic is programmable Cretile which can be programmed to perform many tasks. E.g. If you want switch ON a Cretile Buzzer for 10 seconds and then switch OFF for 5 seconds. And let's say you want to keep doing this over and over. You can instruct Cretile ProLogic to do this job. Writing these instructions for ProLogic is called programming. As explained earlier it can be programmed using software called Ardublock or Arduino IDE as shown earlier in ProLOGIC

ProLogic has inputs, outputs, motor control and wireless communication with Cretile Receiver & Transmitter. All these features can be used in ProLogic program to perform actions according to your project requirement. It has a micro-USB port to which micro-USB data cable can be connected. Other side of micro-USB cable can be connected to PC with Windows OS as shown below image.

Arduino IDE can be used to Program ProLogic. ProLogic can be also be connected any Android Smart Phone with USB-OTG capability using a USB-OTG cable or adapter. Android application can be developed to control ProLogic and communicate with it.

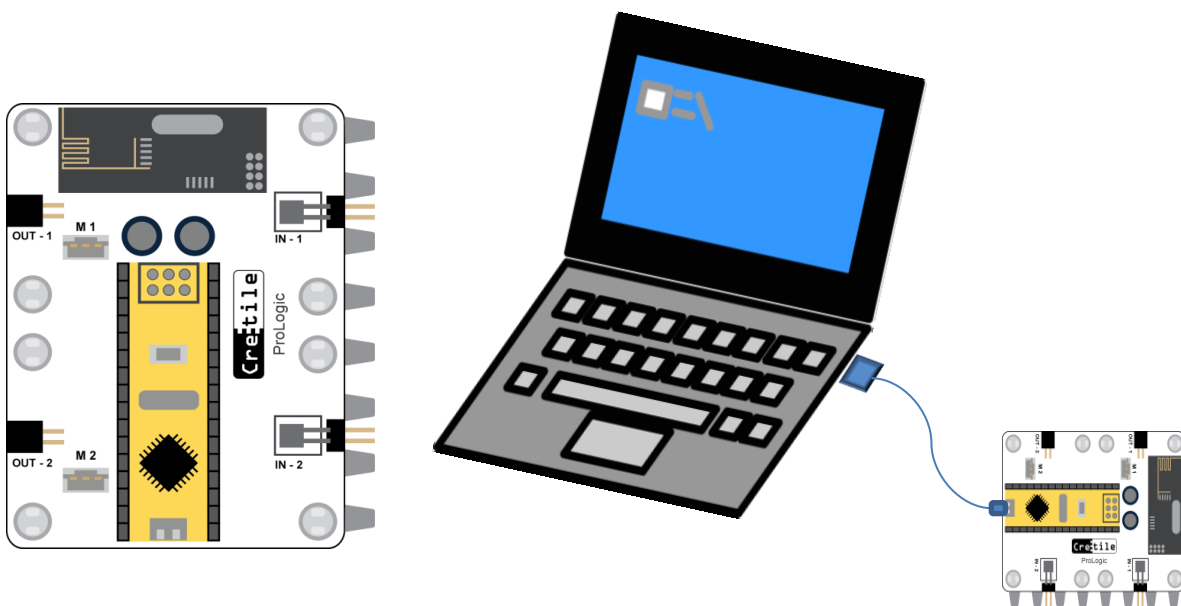


Image 1

4. ProLogic – Software Installation Guide

Installing Arduino software for Cretile

Please download Cretile Software (Cretile.zip) file from link received by you from Cretile Support or Downloads section in <https://www.cretile.com/blog/download-software-and-documentation>. Unzip this in to a folder of your choice preferably to a path without any space in name. After unzipping you will find following directory structure in folder.

Note: if you have received USB drive with Cretile software and documentation then these folders will be present in **<your USB drive>/Cretile Software for ProLogic and Android App Dev.**

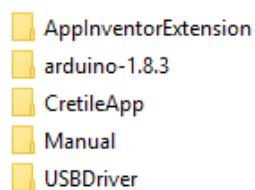


Image 2

In folder Arduino-1.8.3 you will find following files. Double click on Arduino.exe to open Arduino for Cretile. For information on how to use Arduino to develop ProLogic programs see section [

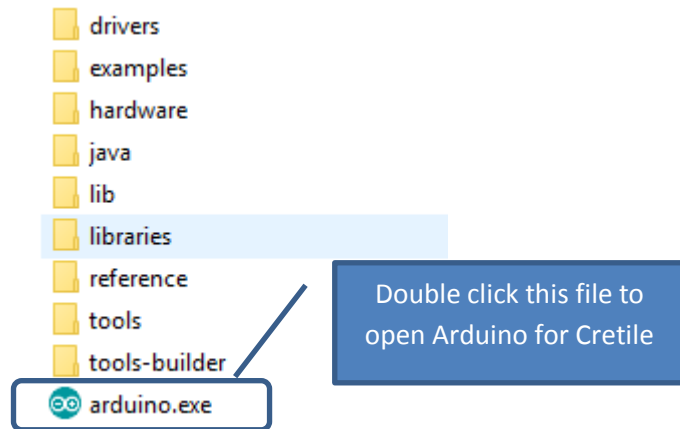


Image 3

You can create a shortcut for this executable by right clicking with mouse on this file and selecting 'Create Shortcut' command. Once shortcut is created you can copy the shortcut to Desktop. You can also select 'Pin to taskbar' command to add it to your taskbar as shown below.

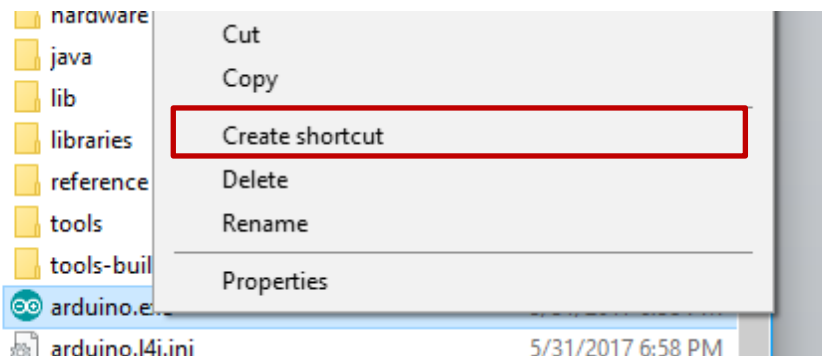


Image 4

Following image show Arduino IDE as it opens when double clicked on Arduino.exe or shortcut created above.

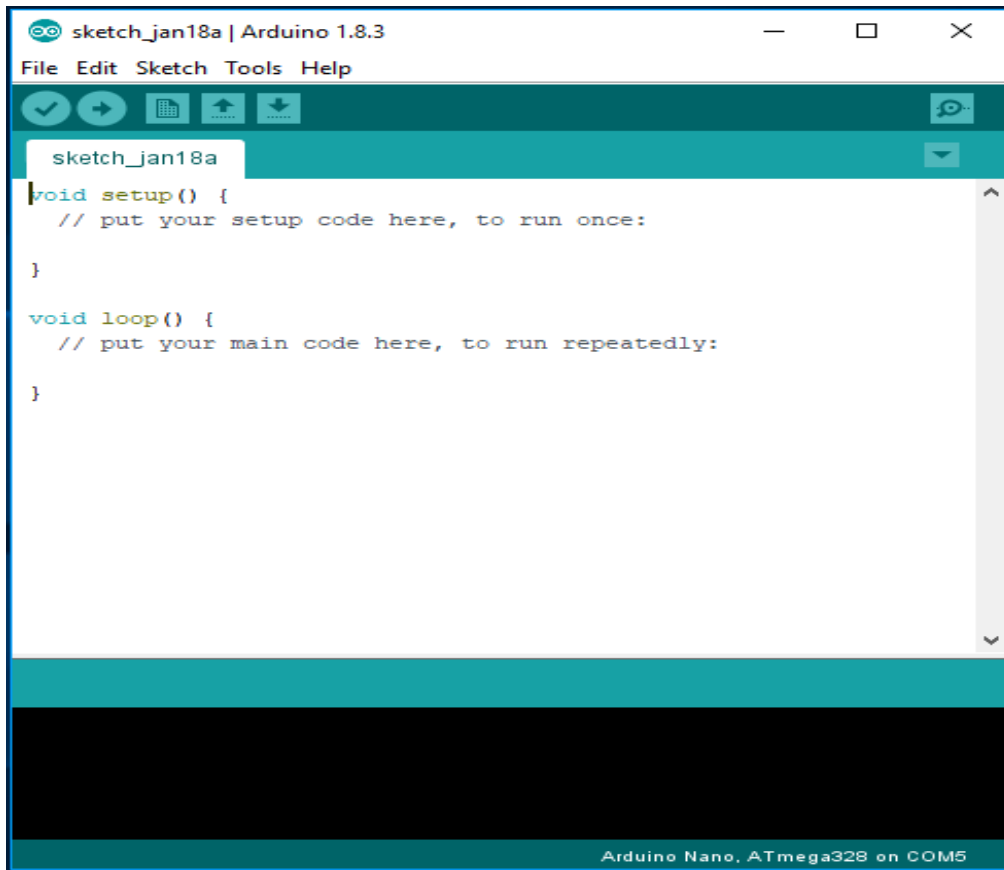


Image 5

Installing USB drivers for Cretile ProLogic

If you are using any Windows version prior to windows 10 (Windows 7, Windows 8) then please install USB driver for Cretile ProLogic. The driver is provided in Cretile.zip file. After unzipping you will find USBDriver folder as shown in section ('Using Cretile Development Software'). In this folder there is SETUP.exe file. Double click on this file to install driver for ProLogic.

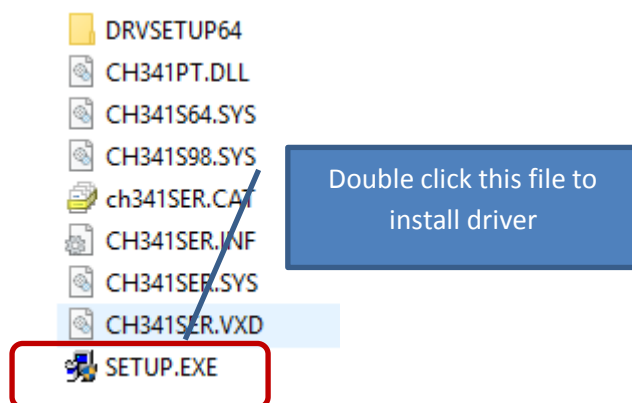


Image 6

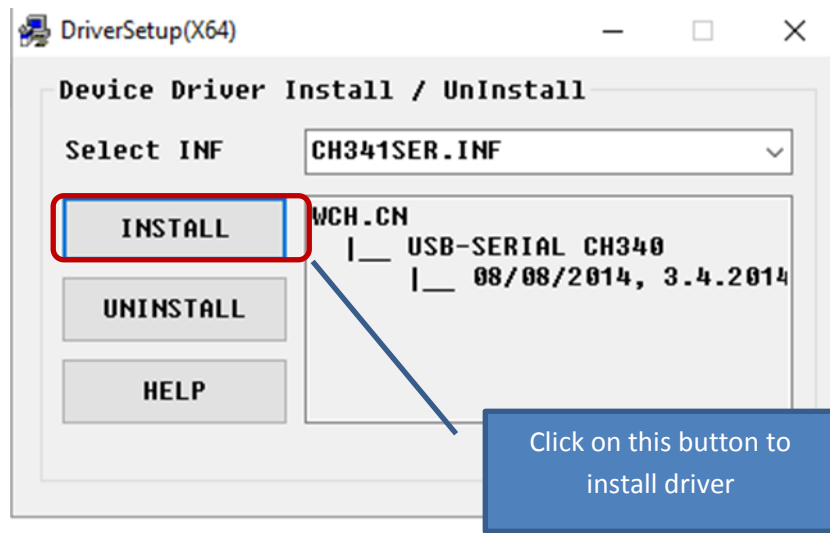


Image 7

Click on install button to install driver.

5. Cretile™ ProLogic – Quick Start Guide

Setting up software for programming

Go to folder *Arduino-1.8.3* (as shown Image 2 on page 38) and double click on *Arduino.exe* to open development environment.

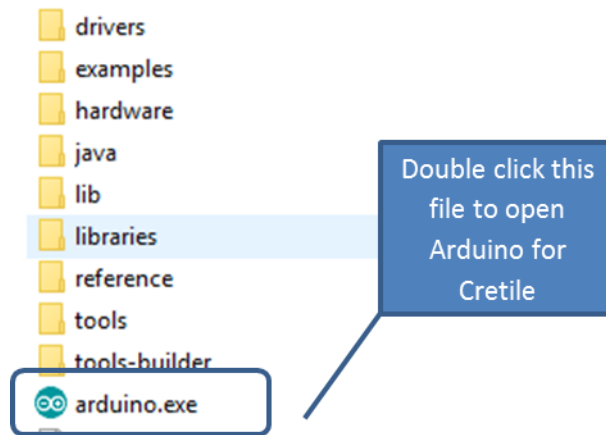


Image 8

This is how the development environment looks like.

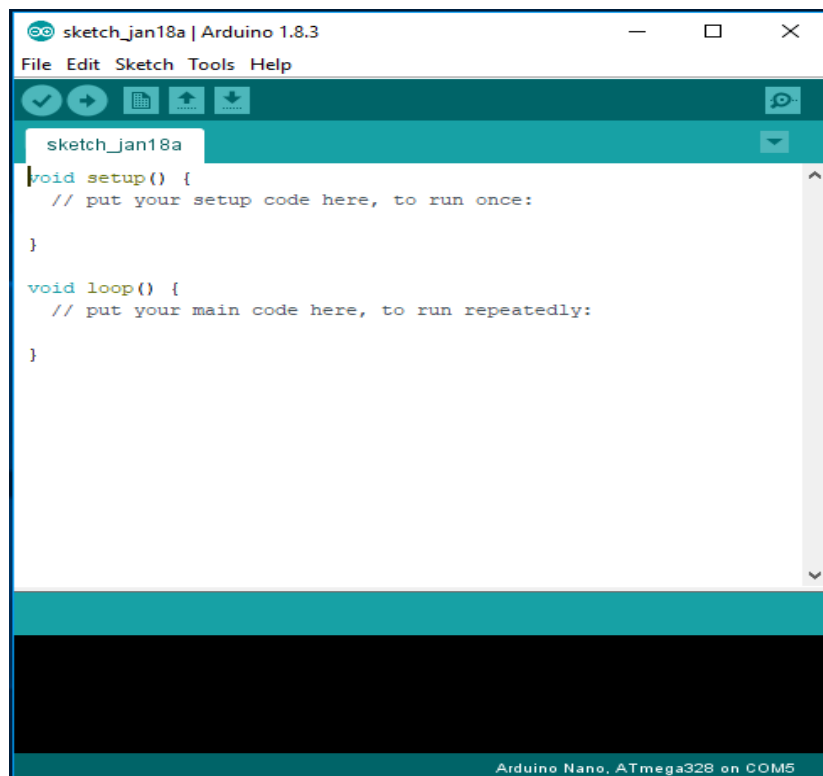


Image 9

Before using this environment do the below setting:

Go to *Tools* menu in Arduino and click on the *Board* sub-menu.

Select “Arduino Nano “from the list.

Now, Arduino environment is configured for use with Cretile ProLogic.

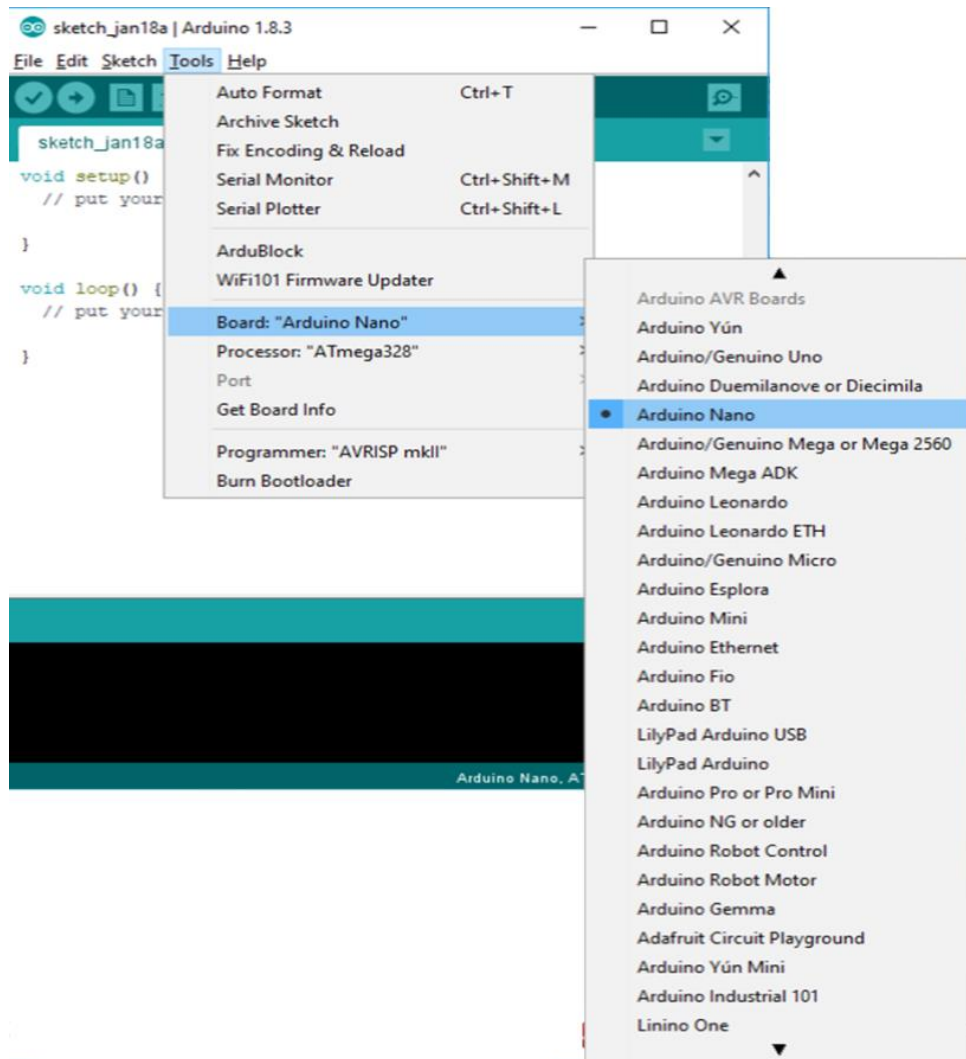


Image 10

Connect ProLogic to your windows computer using micro-USB data cable.

Once connected choose the communication channel via which your computer will talk to ProLogic. To do this, go to *Tools* menu. Choose *Port* sub-menu and choose the listed COM port e.g. COM5 as shown below.

This completes the set-up.

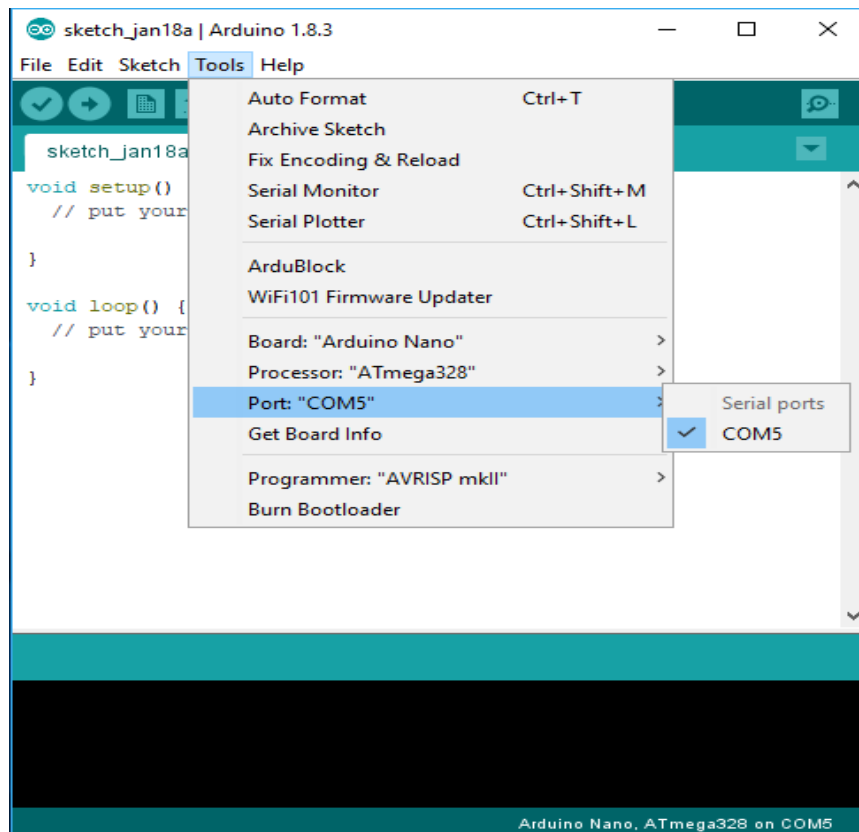


Image 11

If you are C/C++ programmer you can continue to use Arduino Development environment for writing programs as per your project needs.

Alternatively you can use ArduBlock for programming ProLogic. To use ArduBlock, use menu item *Tools* and click on *ArduBlock*.

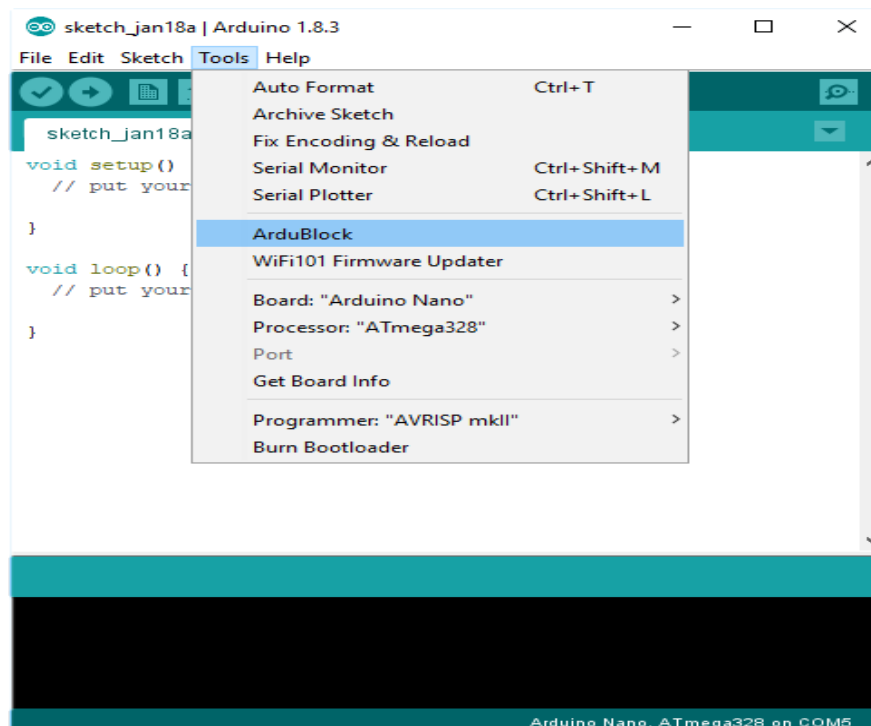


Image 12

ArduBlock tool will appear as shown in below image. Click on *Cretile ProLogic* Drawer and you will find Cretile Blocks supported.

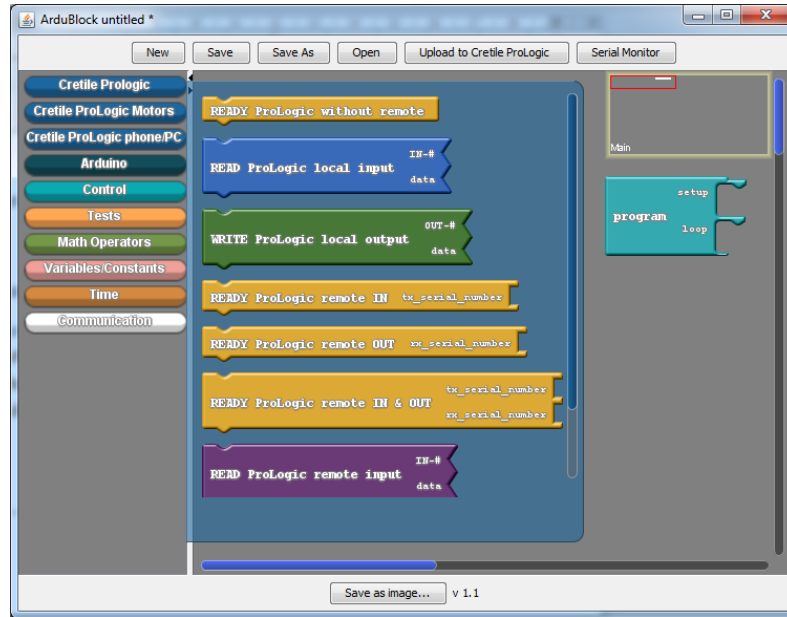


Image 13

Introduction to ArduBlock

ArduBlock has many *block drawers*. The *block drawers* are marked with red rectangle in below image. Cretille ProLogic, Cretille Motors, Cretille Mobile/PC communication are Cretille Block drawers. Each block drawer has set of command blocks. Every command block performs certain function. Information and help of each block can be seen by hovering mouse on block for couple of seconds. Help on the block will appear as shown below.

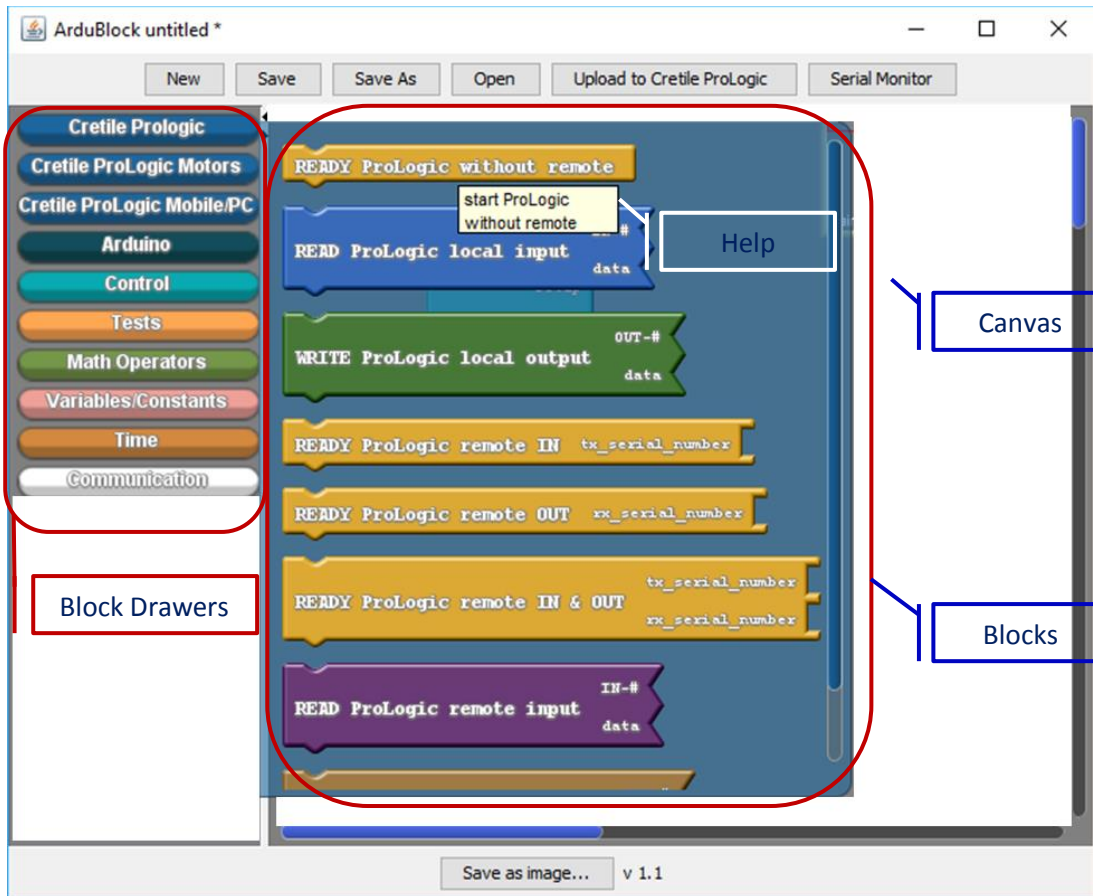


Image 14

On the right side of block drawers section is ArduBlock canvas where blocks can be dragged and dropped to develop a program. Canvas has already a block for an empty program which does nothing. We will add block to make ProLogic ready without remote connection. For this you need to click on Cretille ProLogic drawer and then click and hold on **READY ProLogic without remote** block. Now drag and drop the block in setup section of program as shown below.

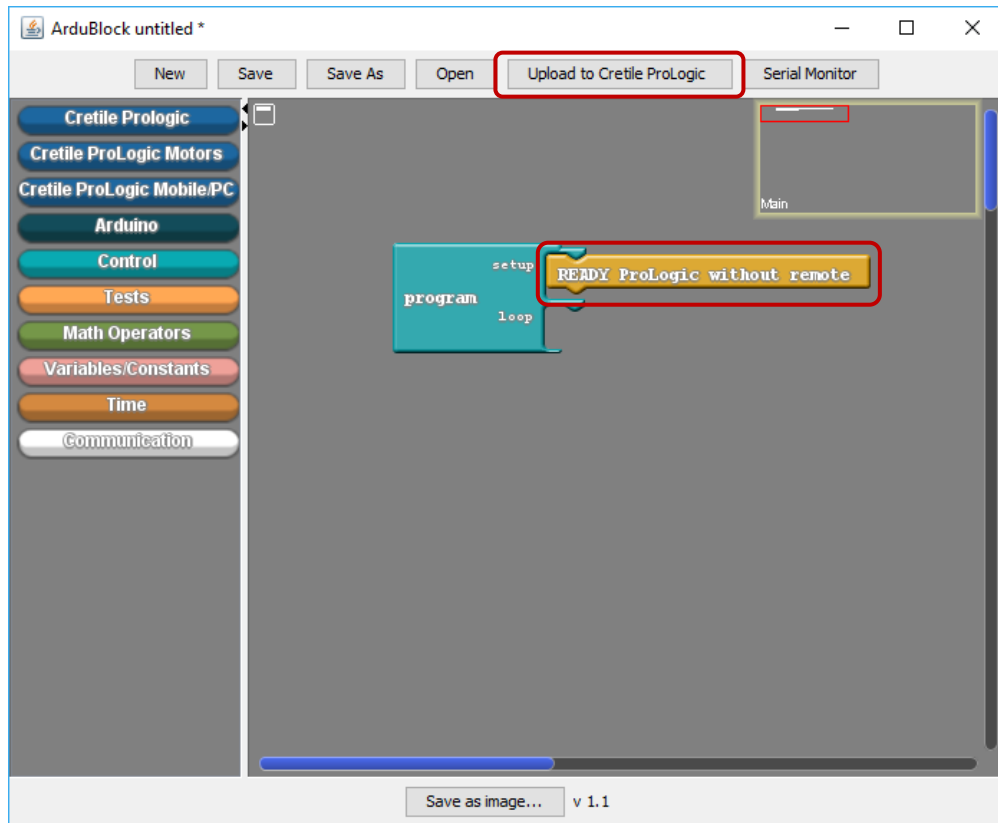


Image 15

You can upload this program to Cretile ProLogic just by clicking on [**Upload to Cretile ProLogic**] button. When you click on [**Upload to Cretile ProLogic**] button, the program you created using ArduBlock is converted to text. A dialog box will also appear to provide name of the file in which generated program can be saved. You can save the program or cancel saving. In any case the program will be compiled and uploaded to ProLogic connected to PC/laptop. Program is generated in C++ and appears in Arduino IDE tool as shown below.

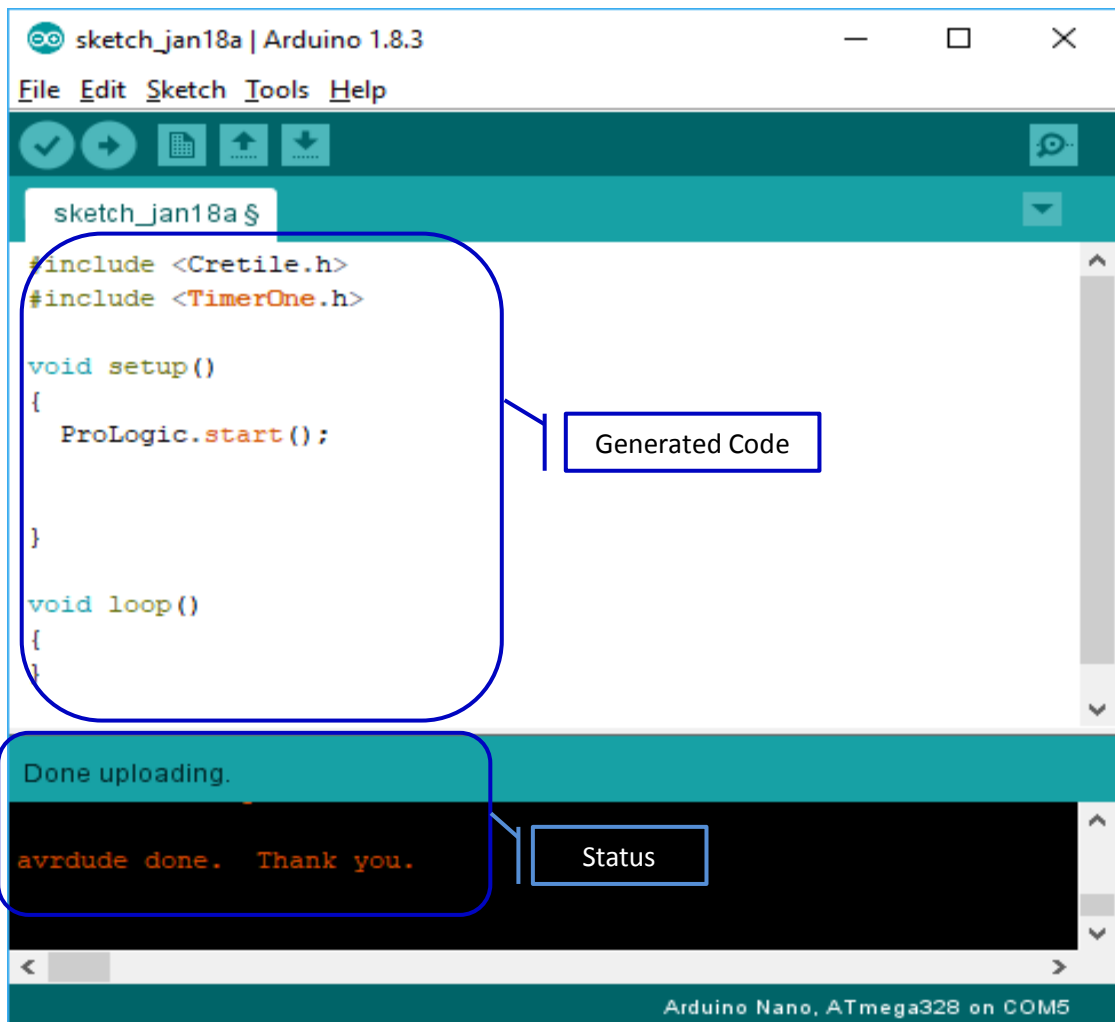


Image 16

Developing first program for Cretile ProLogic

Cretiles Needed

1. Cretile Battery or Cretile Power
2. Cretile Light
3. Cretile ProLogic
4. A micro-USB Cable

Now make connections as shown

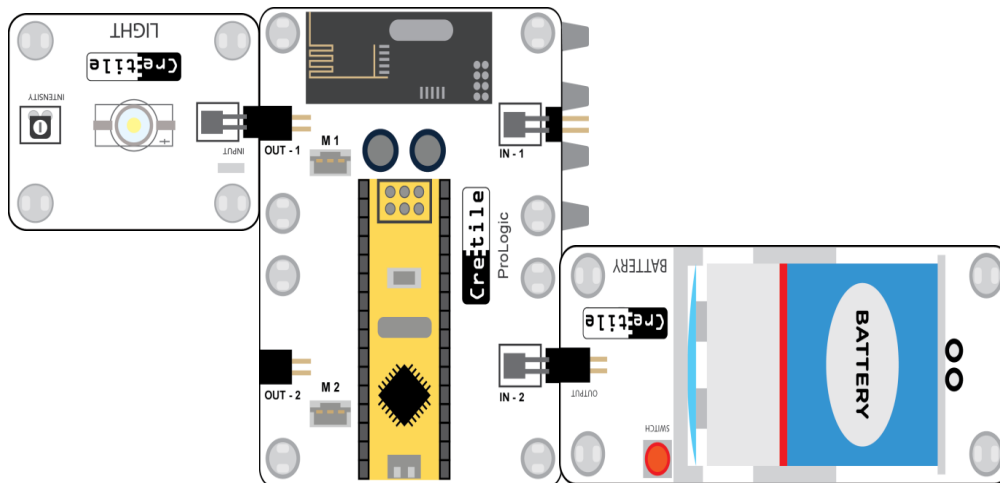


Image 17

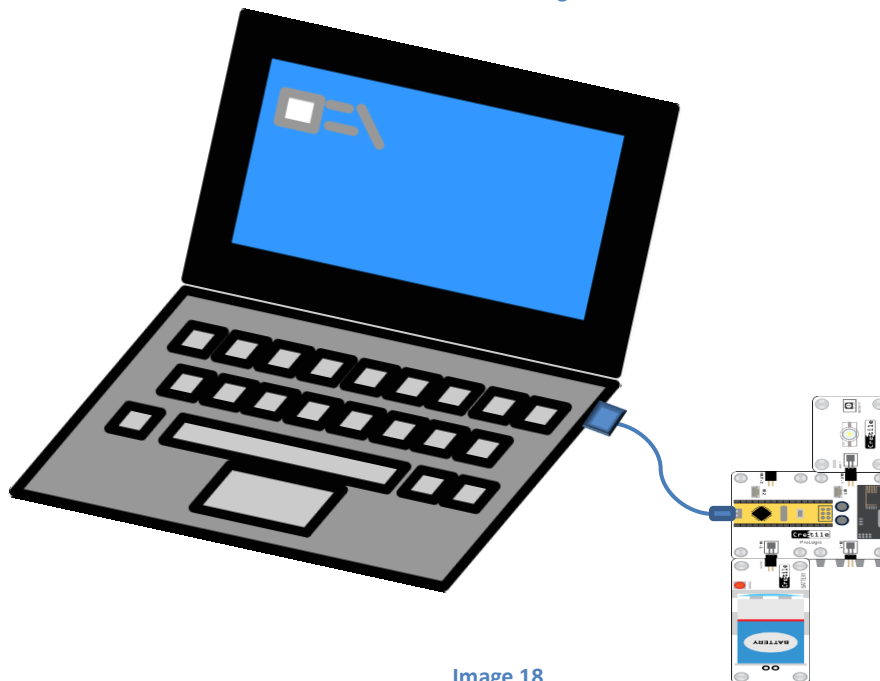


Image 18

Logic

- Step 1. Switch ON Cretile Light
- Step 2. Wait for 1 Sec
- Step 3. Switch OFF Cretile Light
- Step 4. Wait for 1 Sec
- Step 5. Start from Step #1 again

So, if you carry our step 1, 2, 3 & 4 again and again, you will observe a Cretile Light starts to blink. Cretile light shall remain ON for one second and OFF for the next one second. This is happen indefinitely.

Writing ArduBlock Blinking Light Program

1. Open Arduino application.
2. Click on 'Tools' menu option & choose 'ArduBlock' option
3. ArduBlock editor opens up as shown here

On the canvas, you would find a 'program' block which has two sections: first one is setup and second one is 'loop'.

The blocks that you put in 'setup' section gets executed only once when ProLogic is powered up. The blocks that you put in 'loop' section are repeated again and again.

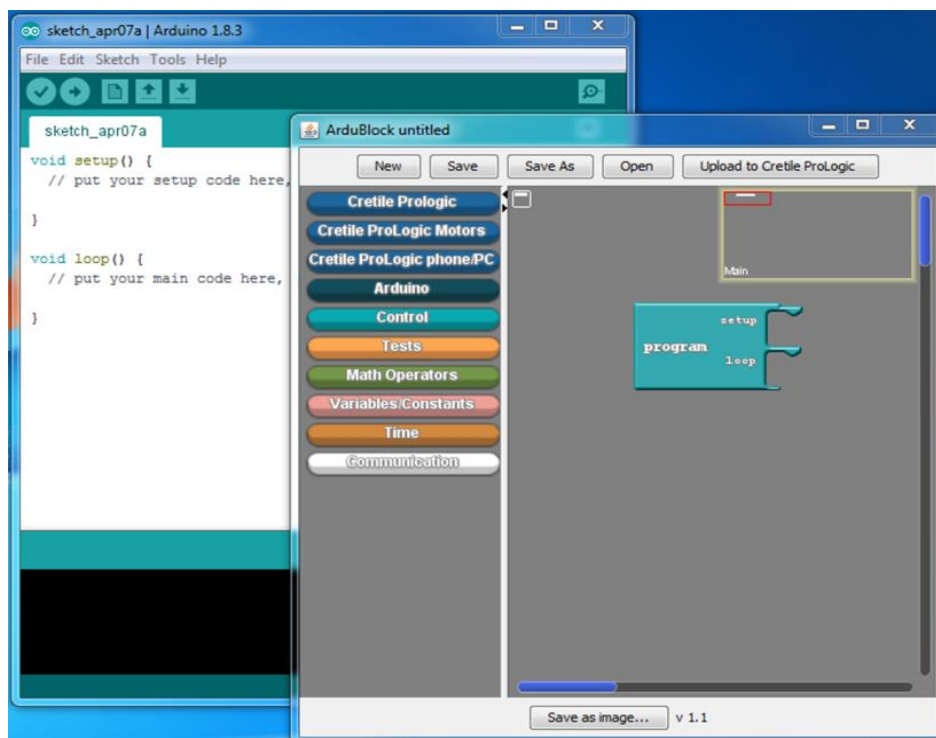


Image 19

Click on 'Cretile ProLogic' drawer from block drawers on the left. You will see all Cretile ProLogic blocks (function) as shown below.

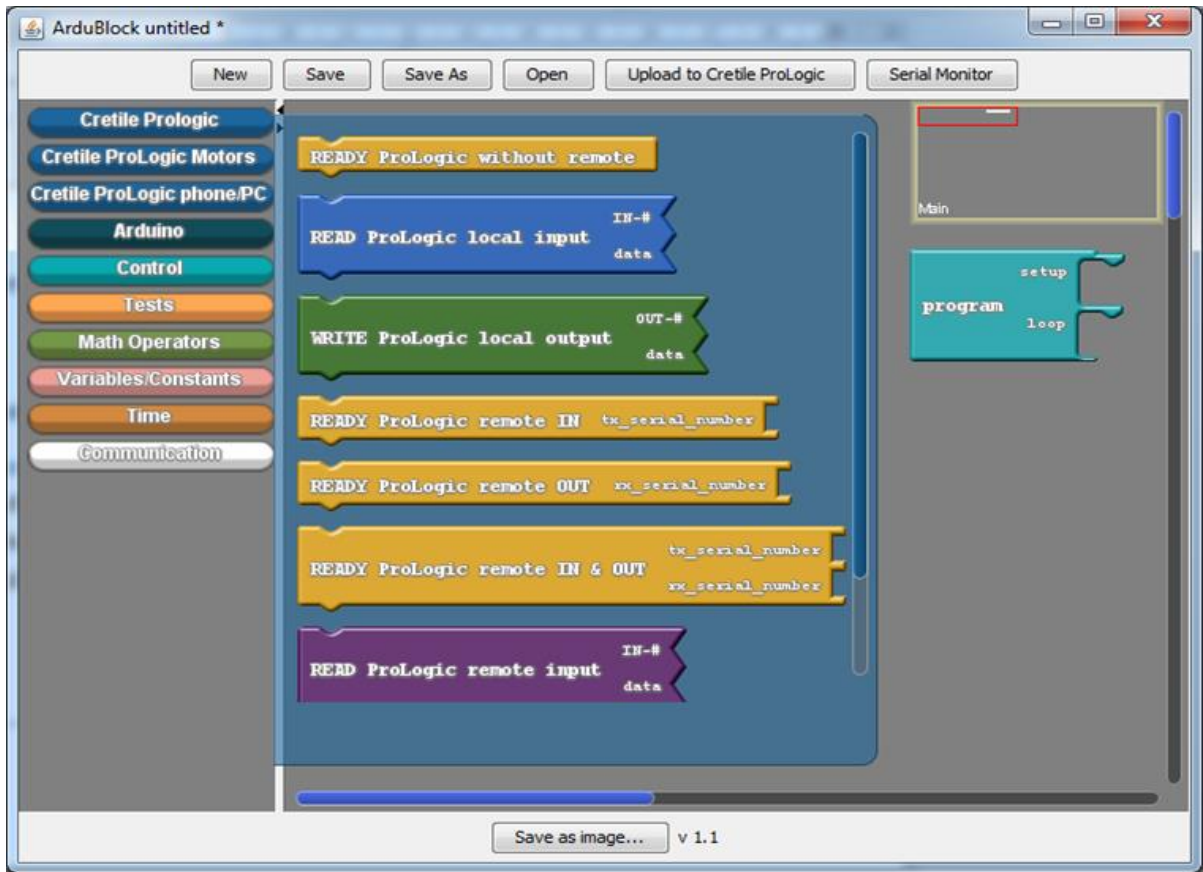


Image 20

Drag *START ProLogic without remote* function and drop it in the *setup* slot of the *program* as shown.

This step makes Cretile ProLogic ready to perform further tasks.

It is like a doctor getting ready by wearing his uniform & gathering all the required equipment for the surgery that needs to be done.

More details are available about this in later chapter.

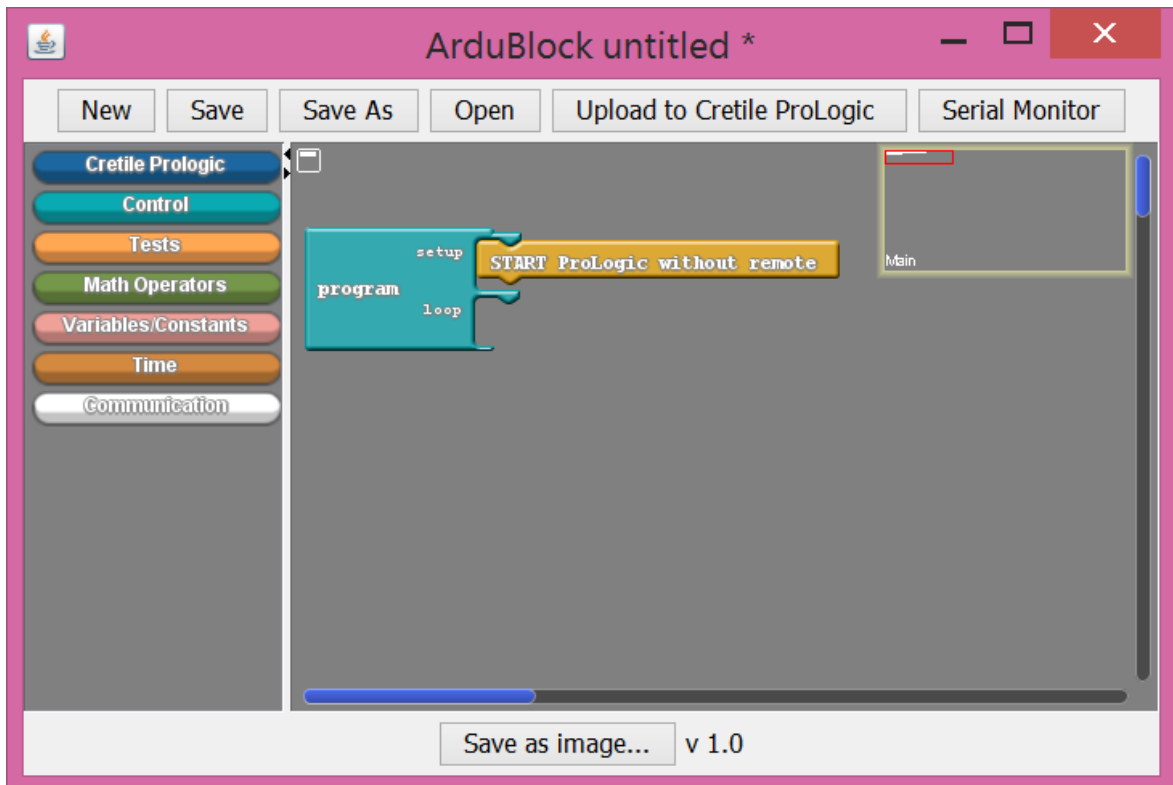


Image 21

Step 1: Switch ON Cretile Light

Drag '*Write ProLogic local output*' and drop it in the loop section. This step will send value '99' to Cretile Light connected at OUT-1.

You can send values from 0 to 99. Zero being lowest and 99 being the highest.

With '99' Cretile Light will be switch ON with full brightness. If value '50' used instead of '99', Cretile Light will be switched ON but with 50% brightness.

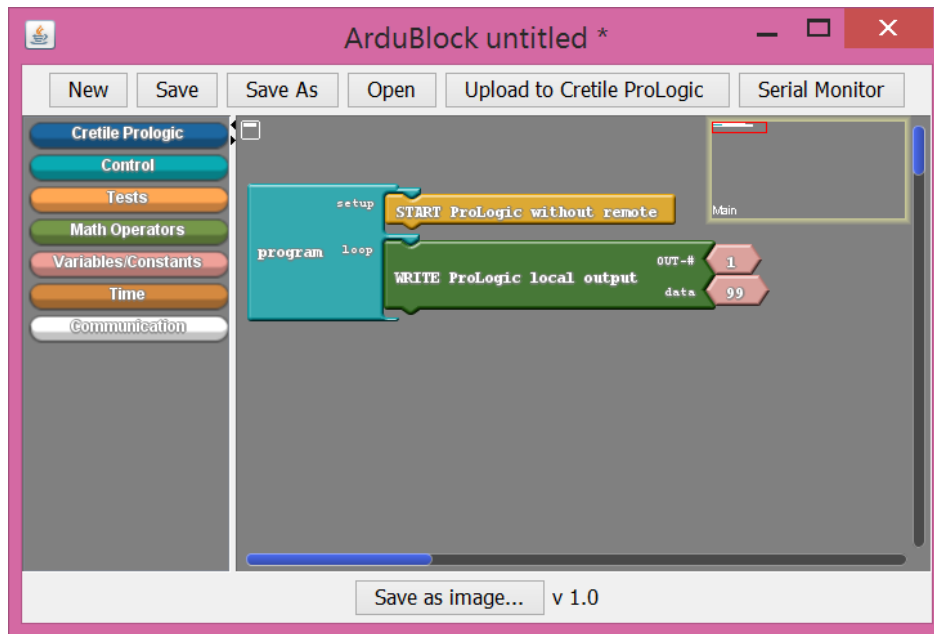


Image 22

Step 2: Wait for 1 Sec

Now drag '*delay SECONDS*' from '*Time*' drawer and drop it after '*WRITE ProLogic local output*' block. This is the second step of our logic in which we wait for one second.

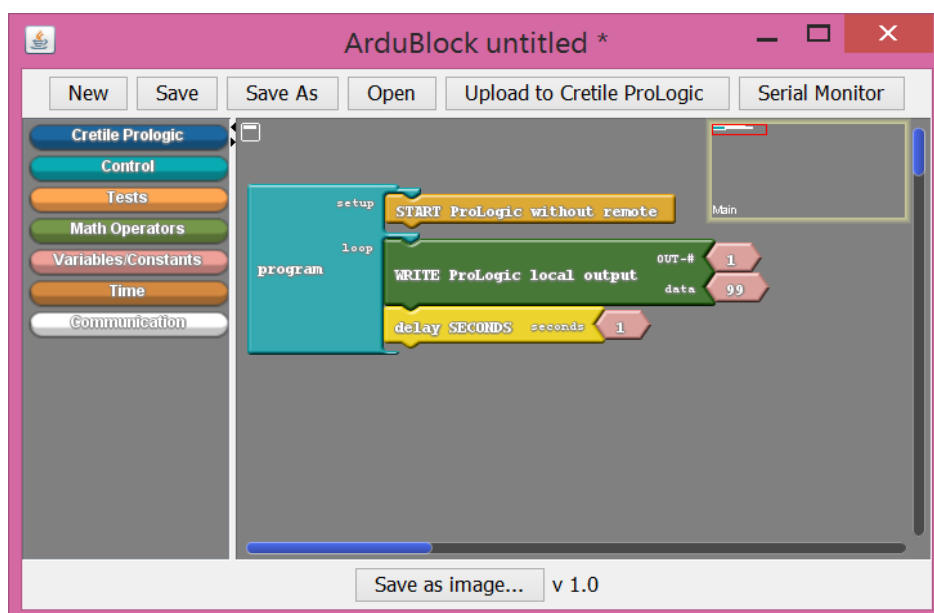


Image 23

Step 3: Switch OFF Cretile Light

Drag '**Write ProLogic local output**' and drop it after the '**delay**'. Click on the **99** and change it to **0**. This will send value **0** to Cretile Light connected at OUT-1 making it switch off.

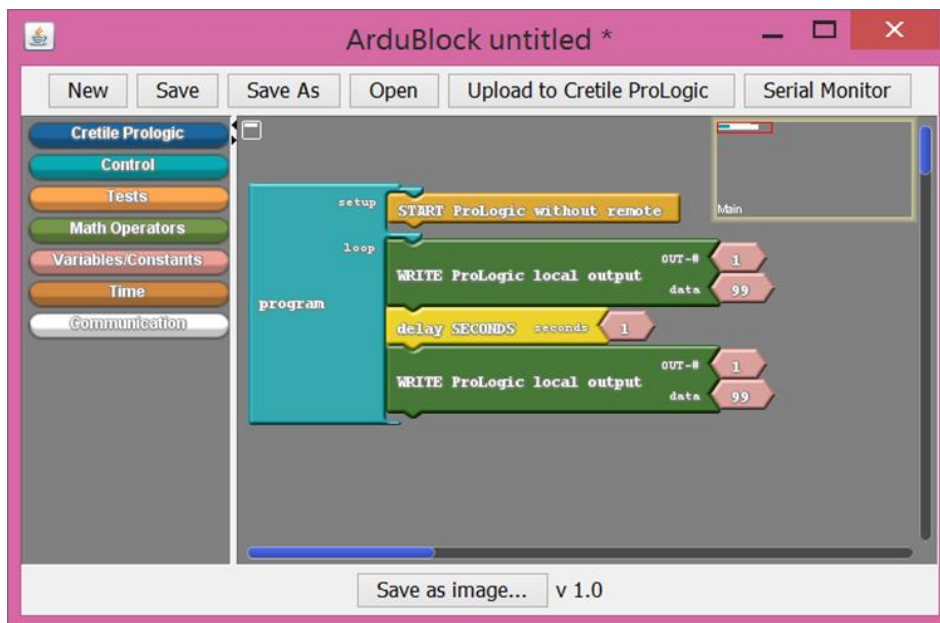


Image 24

Step 4: Wait for 1 Sec

Now drag **delay SECONDS** from **Time** and drop it after second **WRITE ProLogic local output** block.

This is fourth step of our logic in which we wait for one second after the Cretile Light is switched OFF

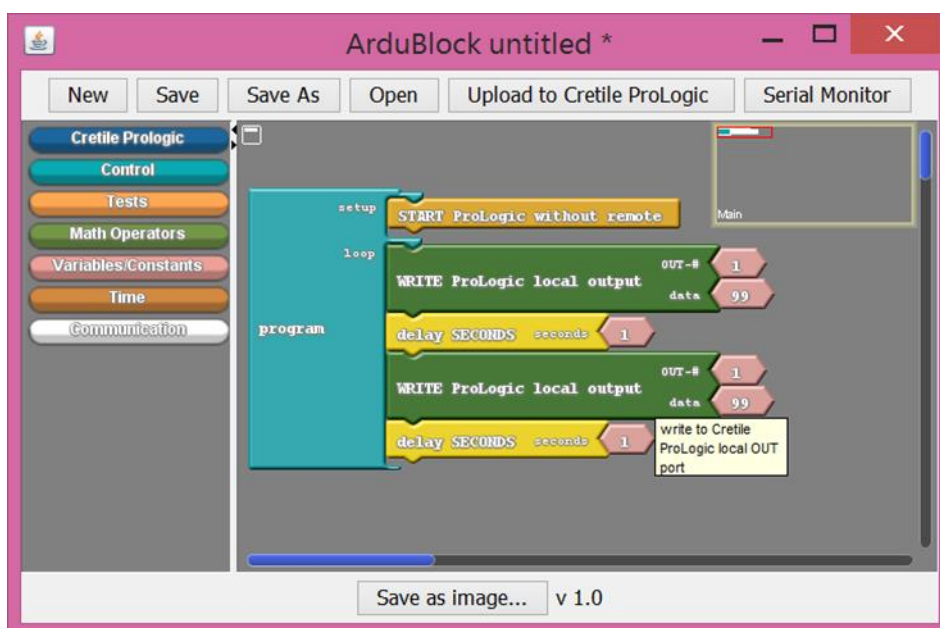


Image 25

Note that all the four instructions are written inside ***loop*** section of ***program***. So, when the fourth instruction is completed, the loop starts again from the first step. This is how the four step logic will get performed again and again until power to ProLogic is switched OFF.

Executing Blinking Light Program on ProLogic

if you want to reuse this program in future, use 'Save' or 'Save As' option from the menu bar and give a suitable name to store it on the computer disk.

You may choose not to save the program on computer disc. You can proceed to next step.

Program we just implemented is present on your computer and needs to be sent (upload) to Cretile ProLogic. Once sent (uploaded), the program will start working on Cretile ProLogic immediately.

To send the program to Cretile ProLogic, click on 'Upload to Cretile ProLogic'.

As soon as the program is successfully sent to Cretile ProLogic the program start to execute and you will find that the Cretile Light connected at OUT-1 starts to blink.

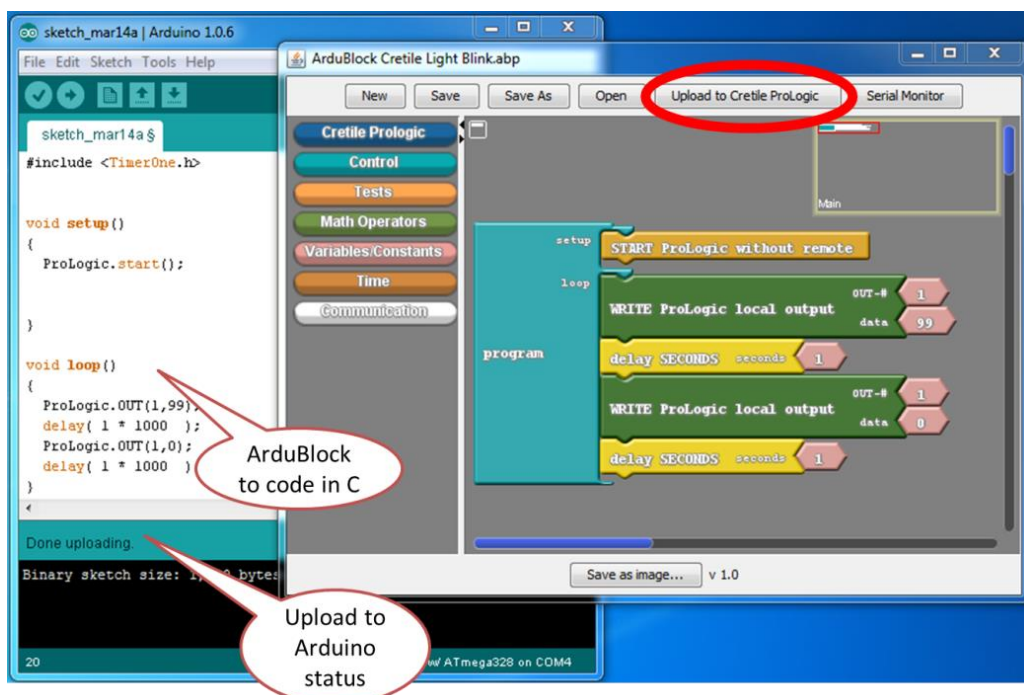


Image 26

Note that 'Upload to ProLogic' step also generates C/C++ code that corresponds to the program implemented using blocks.

You may now remove the USB cable connected between your computer and Cretile ProLogic. The power will be taken from the Cretile Battery and your program will continue to run.

6. ProLogic in Detail

Let's now understand Cretile ProLogic in detail with all capabilities and how to use these capabilities in Cretile ProLogic program.

Cretille ProLogic Capabilities

Inputs and Outputs

ProLogic has two inputs (IN-1, IN-2) and two outputs (OUT-1, OUT-2). Any value from 0 to 99 can be set on OUTputs. When you read INput values in ProLogic program it can have value from 0 to 99. After connecting ProLogic with PC using micro-usb cable as described above, program can be uploaded using Arduino IDE.

In a ProLogic program, you can set the values at OUT ports and read from IN ports by using appropriate commands. See section [[Working with ProLogic local inputs and outputs](#)] for more details.

Motor Control

ProLogic Advanced has built in Motor Controller and two DC motors can be connected and its speed and direction can be controlled using instructions in ProLogic program. See section [[Using ProLogic motors](#)] for more details.

Wireless Communication

ProLogic advanced has capability to communicate with Cretille Receiver and Transmitter and it can be programmed to read INPUTs from Cretille Transmitter and write outputs of Cretille Receiver using appropriate commands described in section [[ProLogic remote inputs and outputs](#)].

ProLogic Development Environment

At the heart of ProLogic is Arduino Nano, which is a popular programmable micro-controller board. Integrated Development Environment software is available to create and upload programs on the board. Cretille ProLogic uses same Arduino IDE for ProLogic program development. In addition to basic IDE there is block based program editor available called ArduBlock. ArduBlock has been customized to add blocks and commands for ProLogic programs. All capabilities of Arduino Nano can be used while developing Programs for Cretille ProLogic with some limitations based on ProLogic commands actually used in programs. For details about Arduino please visit Arduino.cc

ProLogic & Arduino Programming using ArduBlock

ProLogic program structure (Setup & Loop)

ProLogic program consist of two parts Setup and Loop. Setup part is executed once at the beginning of program.

Setup part is used to give commands which are required to make parts of ProLogic ready for its work. You can enable or disable portions of ProLogic capabilities using these setup commands. E.g. if you don't want to use Wireless communication functionality of ProLogic then you can put command in setup section which will only enable local INPUTs and OUTPUTs for use in rest of program. Another example of setup command will be when you want to use wireless communication capability, you need to tell serial number of receiver and transmitter for establishing connection.

Once setup portion of program is executed, commands used in Loop part of program are repeated forever in the sequence in which they are placed in Loop part of the program. E.g. when you want to update OUTput of ProLogic then command to write value of OUTput can be placed in loop and value can change every time the command is executed based on whatever value is supplied to the block. Another example is reading INput values from local and remote inputs. You will get latest value of inputs every time read local or remote IN block is executed.

Let's write a simple program which will make

- ProLogic ready without remote
- Write value 99 on local output OUT-1 & Repeat every 1 second.

As explained earlier, block to make ProLogic ready without remote has to be placed in setup section of program. Block to write value on local OUT-1 will go in loop section. We want to perform this write in every 1 second so we will use Delay block of 1 second inside loop section to wait 1 second. The complete program is shown below.

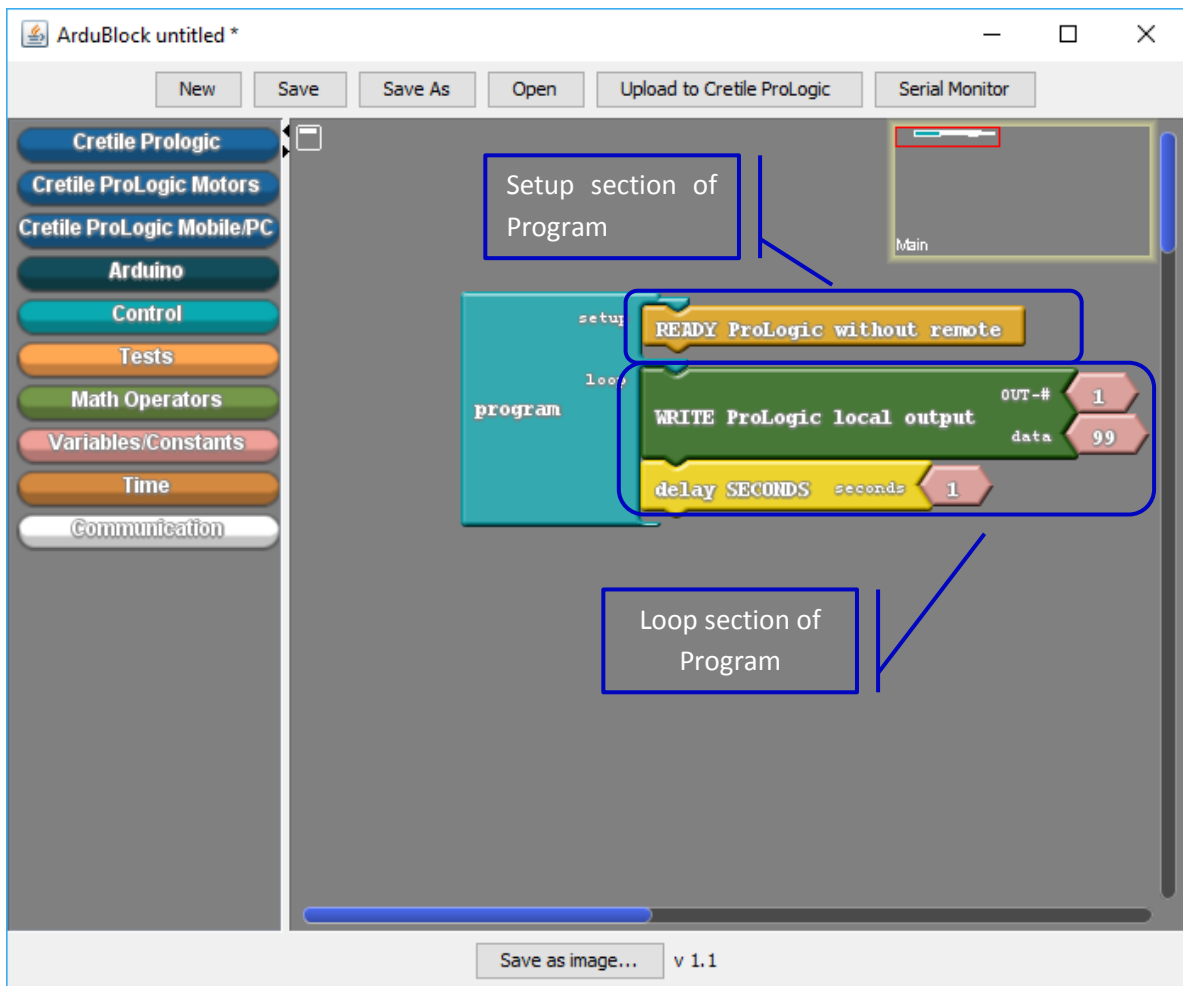


Image 27

Note: If you are familiar with Arduino Programming or Programming in general already, you may skip to section [[Details of ProLogic blocks in ArduBlock](#)]

Control structure

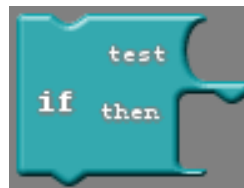
a. *If then, if then else*

When you want ProLogic to do one set of commands under a condition and other set of commands otherwise you can use **if then else** block to do that.

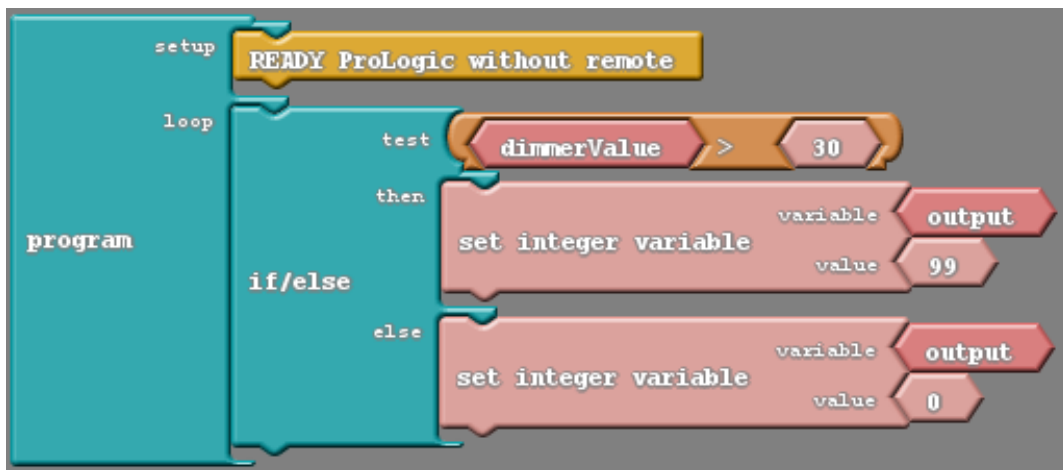
if then else block has condition test, if the condition is **true** then **then** section of commands are executed otherwise **else** section of commands are executed.

You can skip **else** portion of **if then else** block and just provide set of commands to be executed if a condition is **true**. In that case block to be used is **if then** block.

'if then else' and **'if then'** blocks appear in ArduBlock as shown in Figure below.



Following Figure shows example of if then else used in program. This if then else block checks value of variable named dimmerValue and if it is greater than 30 it set value of variable output to 99 else value of variable output is set to 0.



b. Loops

When you want to repeat commands for certain number of time e.g. five times or repeat commands until certain condition occurs, then these loop type of control blocks can be used. Example of loop control blocks are for, while and repeat.

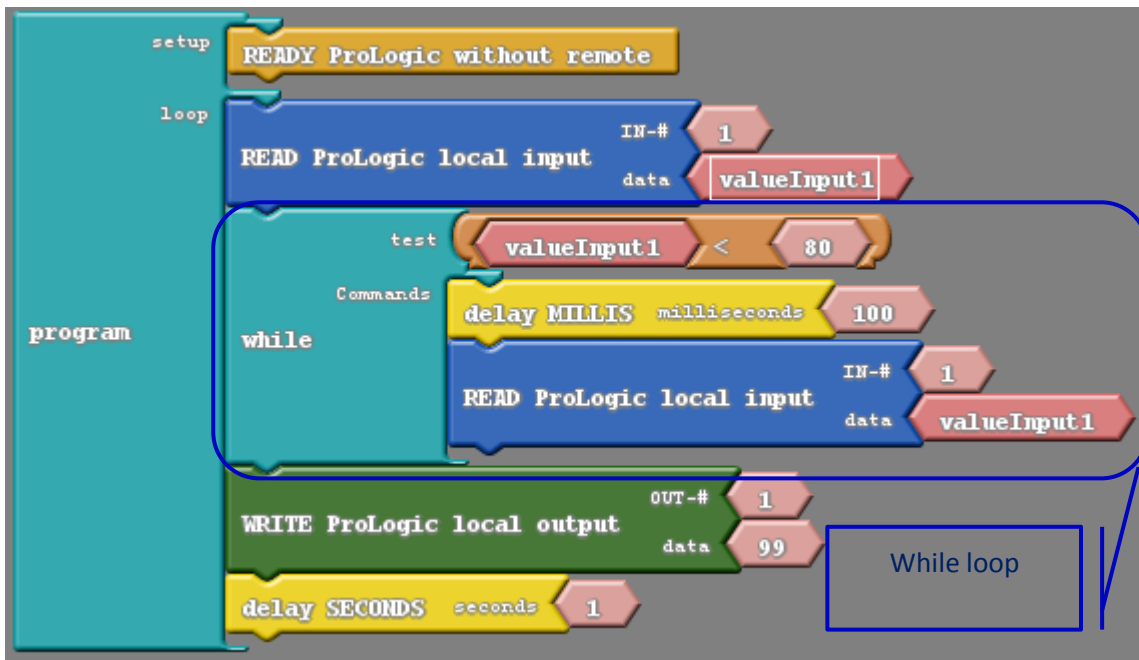
c. While Loops

'While' type of loops have a 'condition' and 'body' of loop. Condition is checked during every iteration of loop.

There are two types of *while* loops in ArduBlock, *while* and *do while*.

While loop first checks condition and then executes commands if condition is satisfied. *do while* loop first executes all commands once and then check condition. So *do while* loop at least executes one time. In case of *while* loop, condition can fail at the start of the loop itself. In that case no commands from body of while loop will execute.

Following Two Figures shows the example of *while* and *do while* loops.



d. Repeat Loops

These types of loops are used to execute commands for certain number of times generally. There are two important types of repeat loop in ArduBlock - 1) *repeat* 2) *repeat and count*.

Repeat loop has two portions, first we need to provide integer value. Second we need to provide commands to be executed in every iteration. Repeat loops executed for number of times as given in integer value.

Repeat and count loop has three portions. First, a variable name which is used to count and you can read its value to get current count. Second portion is integer value, which tells number of times loop commands should be executed. Third portion is commands which need to be executed in every iteration of loop.

Following Two Figures will show the example of *repeat* and *repeat and count* loops.

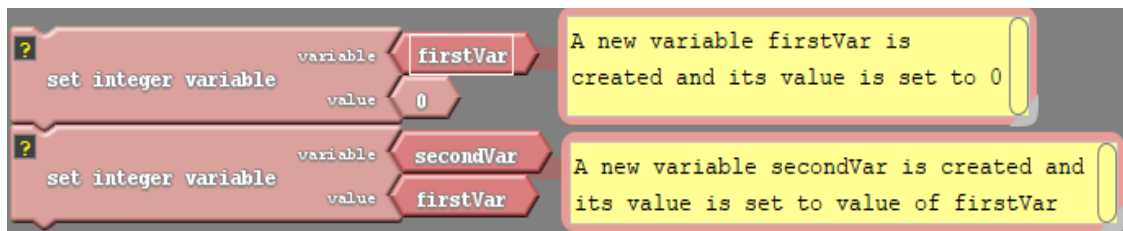


Variables

Variables are place holders or memory of a specific type of value like number or text. They are symbols which can remember value of specific type. A number variable can remember any value between 1 to N.

Value of variable can be set by various commands e.g. command to read ProLogic input can set value of variable with whatever is the value of ProLogic input. You can also set value of a number variable to any number for example 10 by using assignment block.

When you want read value of variable remembered by it you can simply place the variable where some number value is required by block e.g. while setting value of one variable to value of some other variable as shown below.



Time functions

Time drawer provides commands that will wait for specified time when executed. E.g. if command Delay Seconds with value 2 is used then it will wait and do nothing during 2 seconds. After 2 seconds, command placed after the delay command will be executed.



Details of ProLogic blocks in ArduBlock

Working with ProLogic local inputs and outputs

As explained earlier, ProLogic has two inputs and two outputs. Current value of inputs can be read by using *READ ProLogic local input* block and outputs can be written using *WRITE ProLogic local output* block.

a. *READY ProLogic without remote*

To use ProLogic feature of controlling local outputs it need to be made ready for that. Ready ProLogic without remote block can be used for that. This block should be put in setup section since this block needs to execute only once. When you make ProLogic ready for remote outputs as described in section [[ProLogic remote inputs and outputs](#)], ProLogic local outputs can also be used without any additional setup.

Ready ProLogic without remote block appears as shown below in ArduBlock .

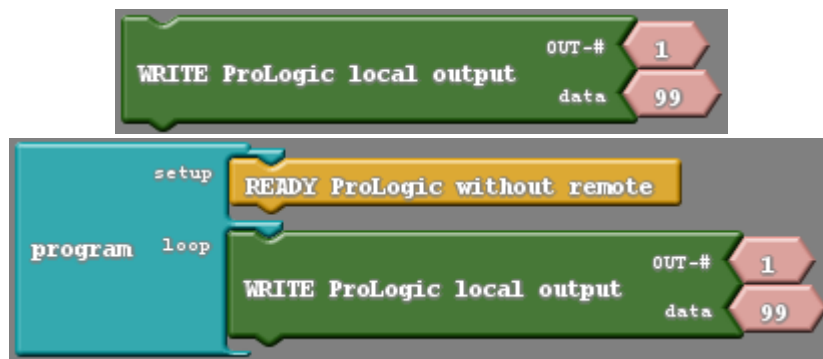


b. *WRITE ProLogic local output*

WRITE ProLogic local output is used to write value on OUT-1 or OUT-2. OUT-# field of the block is used to specify port number. For OUT-1 it will be 1 and for OUT-2 it will be 2.

Any number from 0 to 99 can be written on OUT-1 or OUT-2 of ProLogic. Number can be a constant number or a number variable.

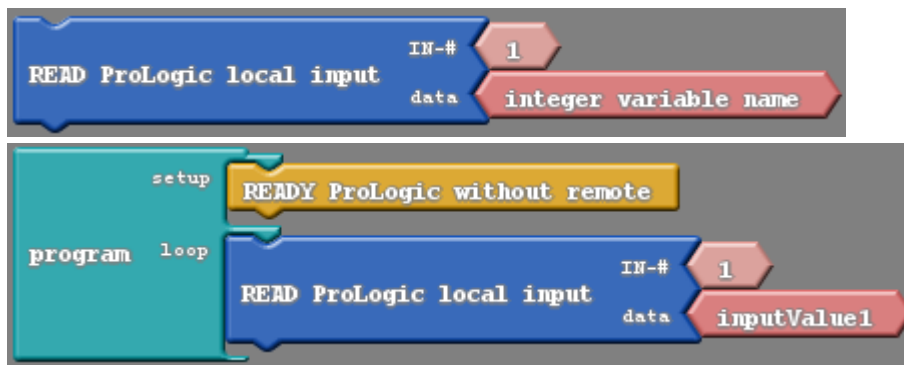
WRITE ProLogic local output appears in ArduBlock as shown below.



c. *READ ProLogic local input*

READ ProLogic local input block is used to read value of IN-1 or IN-2 and remember it in a variable. INput values can be number from 0 to 99.

READ ProLogic local input block appears in ArduBlock as shown below.



Examples: Local inputs and outputs

Let's use these blocks to develop two simple programs

1. Blink

You will develop a program which will blink a light connected to ProLogic OUT-1. For this program, you need Battery, ProLogic and Light module. Connect these modules as show below.

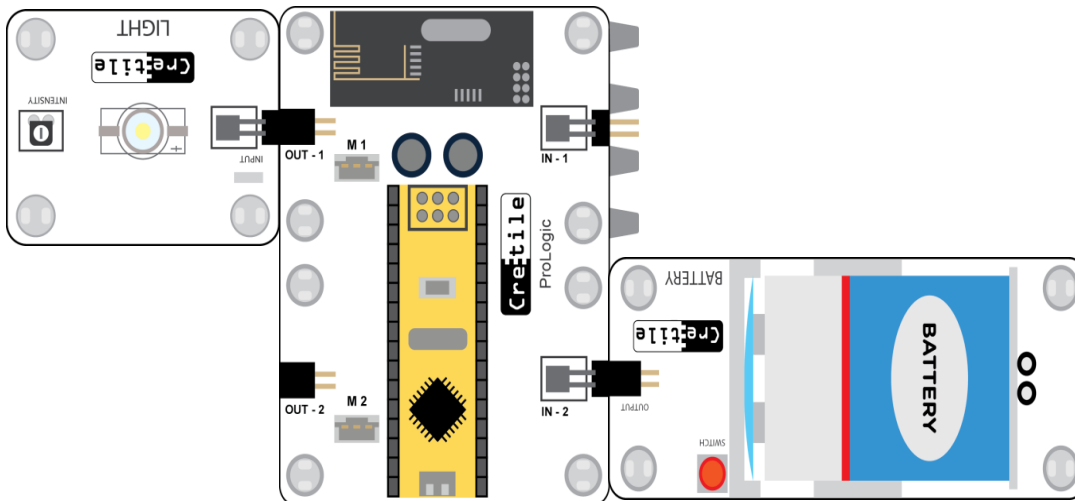


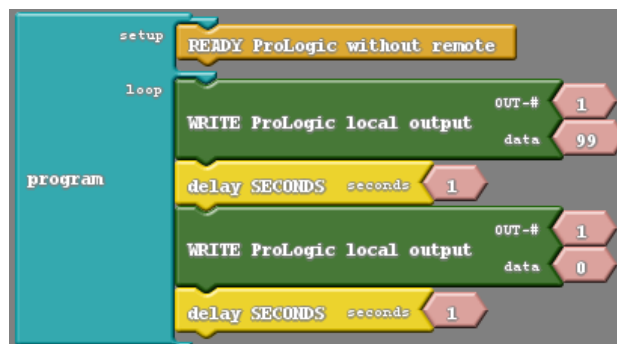
Image 28

To develop program, open ArduBlock. In this program you don't need wireless communication and motors so you should make ProLogic ready without remote in setup section of program. You don't have to use **READY ProLogic MOTORS** block since you are not going use motors in this program.

To blink a Light every 1 second indefinitely you need to switch ON light for 1 second and switch OFF light for 1 second and repeat these commands indefinite.

- Cretile Light is connected on OUT-1 so you need to switch on OUT-1 for 1 second and switch off OUT-1 for 1 second and repeat this indefinitely.
- To switch on or off OUT-1 you need to use **WRITE ProLogic local output** block. To switch on Light fully at maximum brightness you need to write 99 to OUT-1.
- After switching on OUT-1, you should wait for 1 second. For this **delay SECONDS** block should be placed after **Write ProLogic local output** block.
- After this **delay SECONDS** block you need to switch off OUT-1 so again you need to use **WRITE ProLogic local output** block and write value 0 on OUT-1.
- Light should be off for 1 second so you need to put **delay SECONDS** block with value 1.
- These commands need to repeat forever so these should be in Loop section of Program.

Complete program is shown in below Figure for blinking Light.

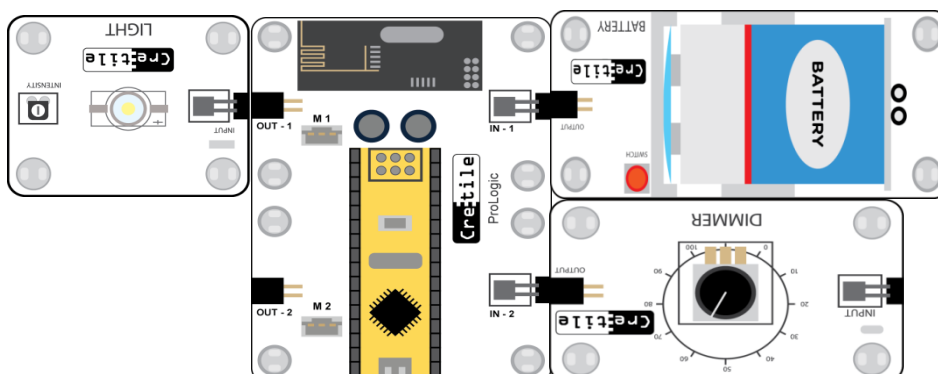


2. Setting speed of blinking using Dimmer and ProLogic

To control the speed of blink you need to change delay time in above program. For faster blinking delay time should be smaller and slower blinking delay time should be higher.

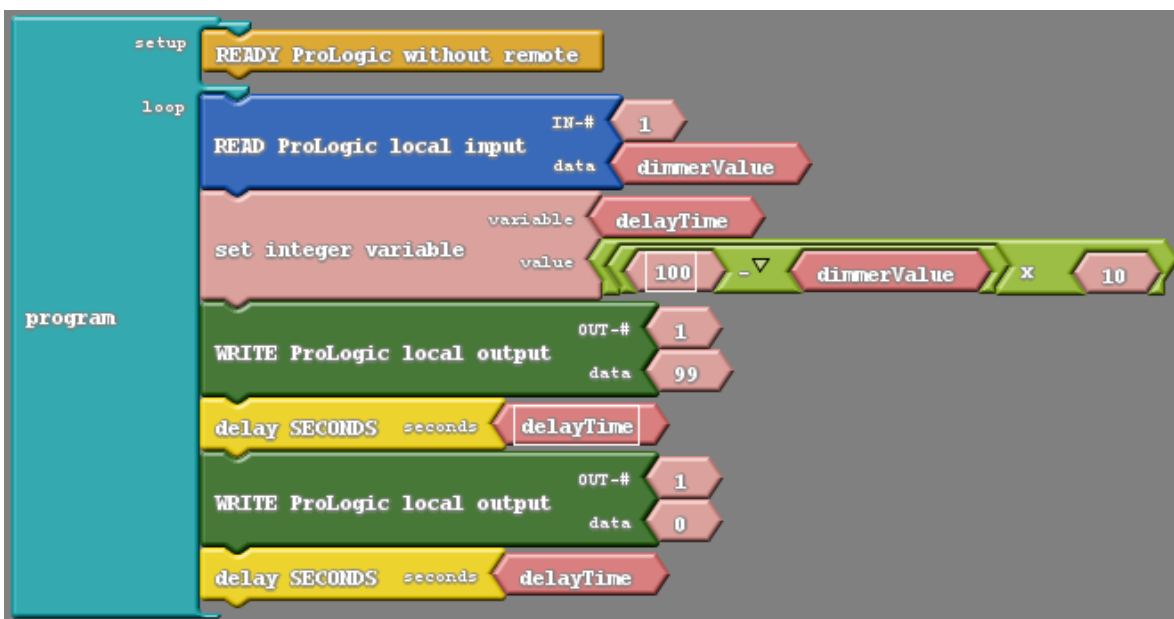
In previous program blink speed and hence delay time was constant. If you want to change blink speed, delay time should change. Actual delay time will be according to value set on Dimmer.

For this program you need Battery, ProLogic, Dimmer and Light module. Connect these modules as shown below.



- Dimmer is connected to **IN-2** of ProLogic. To read Dimmer value (from 0 to 99) you need to use Read ProLogic IN block. Actual value is remembered in a variable. You have named the variable as dimmerValue
- Delay time for on and off for **OUT-1** need to be calculated using dimmerValue variable after reading dimmerValue from IN-1.
- You will use **delay MILLIS** block to keep OUT-1 ON or OFF for calculated duration. For this you need to calculate delay time in milliseconds. When dimmer Value changes from 0 to 99, delay time should change from 1000 milliseconds to 10 milliseconds. For this formula is $\text{delayTime} = (100 - \text{dimmerValue}) * 10$
- After calculating delayTime you can use delayTime variable in **delay MILLIS** block to change delay time for ON and OFF actions

Complete program is shown in below figure.



ProLogic remote inputs and outputs

ProLogic advanced can communicate with Receiver and Transmitter. It can write to receiver outputs OUT-1 to OUT-4 using **WRITE ProLogic remote output** block. It can read from Transmitter inputs IN-1 to IN-4 using **READ ProLogic remote input** block.

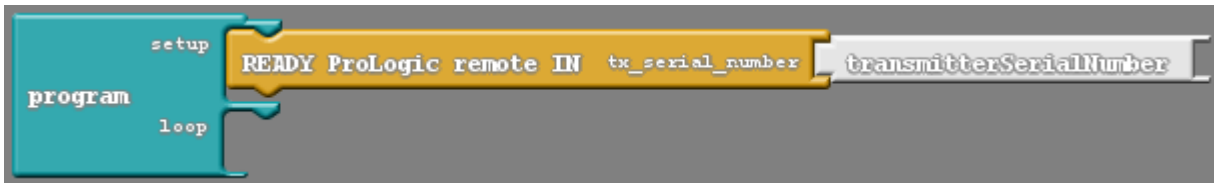
These blocks are very similar to reading and writing local blocks as described above.

a. *Ready ProLogic remote IN*

When you want to use only remote inputs using Transmitter then **READY ProLogic remote IN** can be used to set it up for that. This block generally is used in setup section of program. You will transmitter serial number as input to this block. You can find Transmitter serial number written at the backside of Transmitter module.

When you use this block, ProLogic local inputs and outputs can also be used in program apart from remote INputs.

Ready ProLogic remote IN block appears in ArduBlock as shown below.

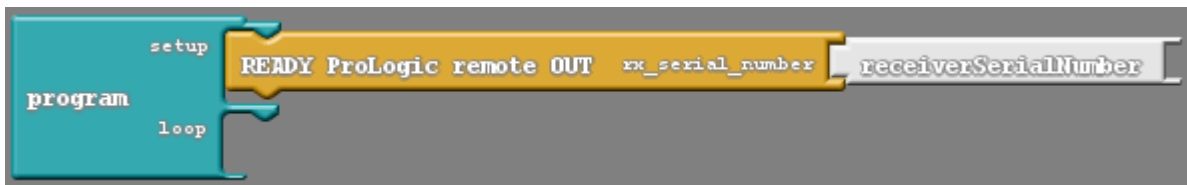


b. *Ready ProLogic remote OUT*

When you want to use only remote outputs using receiver then **READY ProLogic remote OUT** can be used to set it up for that. This block generally is used in setup section of program. You will need receiver serial number as input to this block. Receiver serial number is written at backside of Receiver module, please type the serial number mentioned there as input rx_serial_number.

When you use this block, ProLogic local inputs and outputs can also be used in program apart from remote outputs.

Ready ProLogic remote OUT block appears in ArduBlock as shown below.



c. *Ready ProLogic remote IN & OUT*

When you want to use remote INputs as well as OUTputs using Transmitter & Receiver module then **READY ProLogic remote IN & OUT** can be used to setup ProLogic. This block generally is used in setup section of program.

When you use this block, ProLogic local inputs and outputs can also be used in program apart from remote OUTputs.

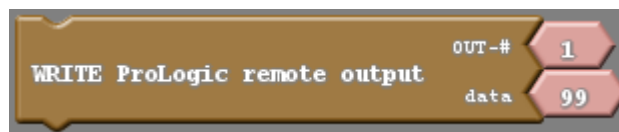
READY ProLogic remote IN & OUT block appears in ArduBlock as shown below.



d. WRITE ProLogic remote output

WRITE ProLogic remote output is used to write value on OUT-1 to OUT-4. Any number between 0 and 99 can be written on OUT-1 or OUT-2 of ProLogic. Number can be a constant number or a number variable.

Write ProLogic remote output block appears in ArduBlock as shown below.



e. READ ProLogic remote IN

READ ProLogic remote input block is used to read value of IN-1 to IN-4 of Transmitter module and remember it in a variable. INPUT values can be number between 0 and 99.

READ ProLogic remote input block appears in ArduBlock as shown in below Figure.



Examples: Remote inputs and outputs

1. Remote vehicle control using program

Let's use remote inputs and outputs to control a wireless vehicle using program in ProLogic. For this you will need 1 Power module, 1 Battery, 1 Receiver, 2 Motors and ProLogic module. Connect these modules as shown in below figures

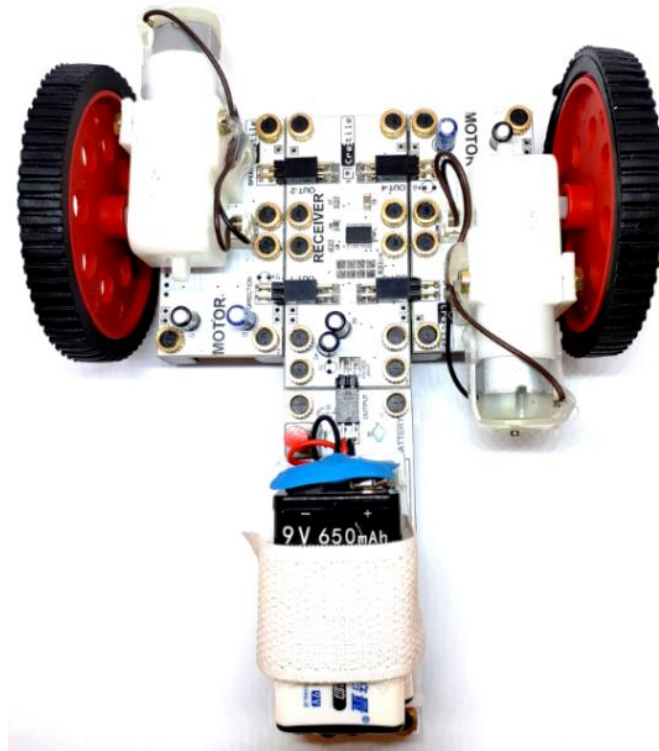


Image 29

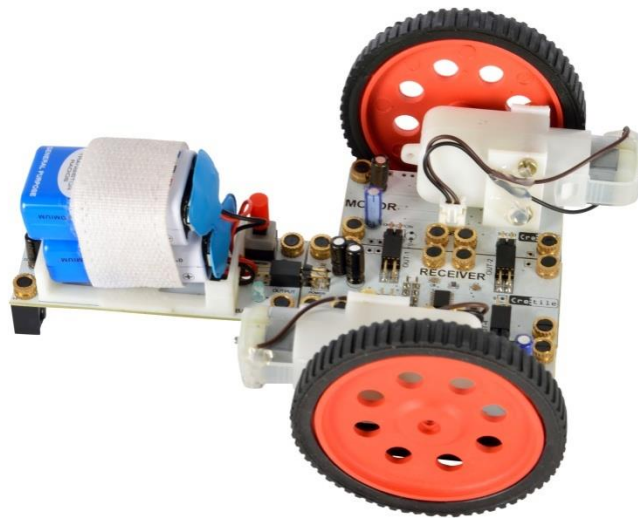


Image 30

Let's write a program which

- Make ProLogic ready for writing remote OUTputs on receiver
- Moves vehicle forward for 3 seconds
- Turns vehicle by 90 degrees and moves forward vehicle for 3 seconds
- Repeats this indefinitely.

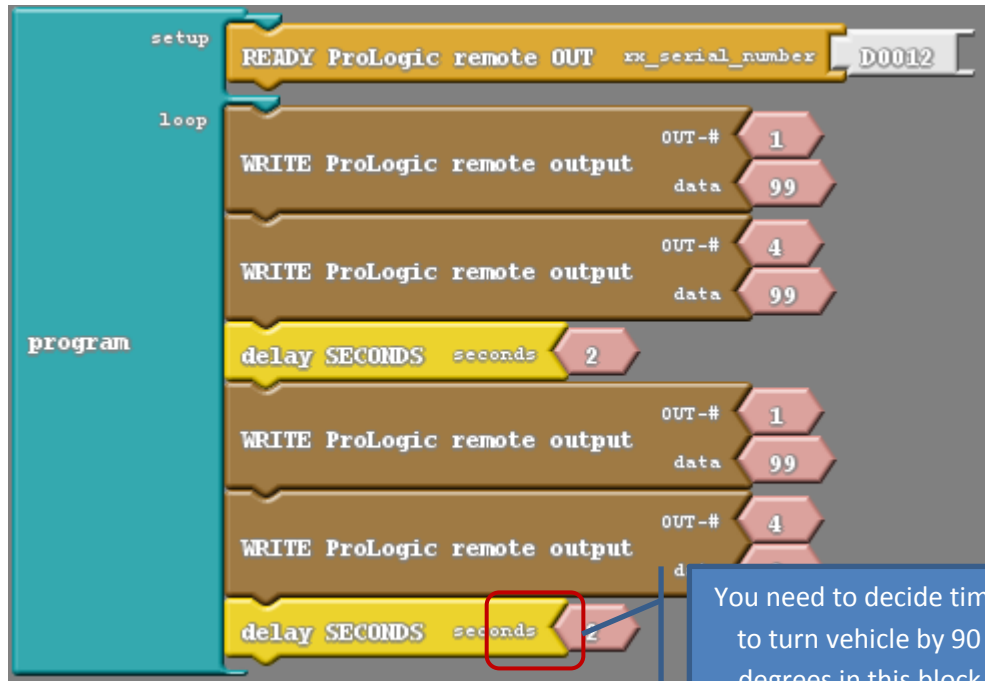
- ***Make ProLogic Ready to control remote OUTputs***
You will need receiver's serial number to initialize ProLogic for communication with receiver and control its OUTputs. Serial number of receiver and transmitter is written at the back of Transmitter and Receiver. Please check and use the serial number mentioned there in block for making ProLogic ready with remote OUT block as described in section [[Ready ProLogic remote OUT](#)].

- ***Move vehicle forward***
 - To move vehicle forward you have to rotate both motors in same direction. Motors are connected to receiver facing opposite direction to each other. Hence to rotate both motors in same direction one of motor direction should be set opposite to other.
 - Speed control of MOTOR-1 connects to OUT-2 of receiver
 - Speed control of MOTOR-2 is connected to OUT-4 of receiver.
 - To turn vehicle you have to write 99 on remote OUT-2 and write 0 on remote OUT-4 for some time.

- ***Turn Vehicle by 90 degrees***
 - To turn vehicle by 90 degrees you have to keep one MOTOR switched ON and other motor OFF for some time. This time need to be decided after experimenting.

- ***Repeat above steps forever***
 - This sequence of moving forward and turning 90 degrees is repeated forever by putting this sequence of commands in Loop part of ProLogic program.

Complete program is shown below.



Using ProLogic motors

ProLogic has capability to directly connect motors without Motor module. There are two sockets where motor pins can be connected. Compatible motors are available at www.cretille.com. Image 31 shows two motors connected to ProLogic board.

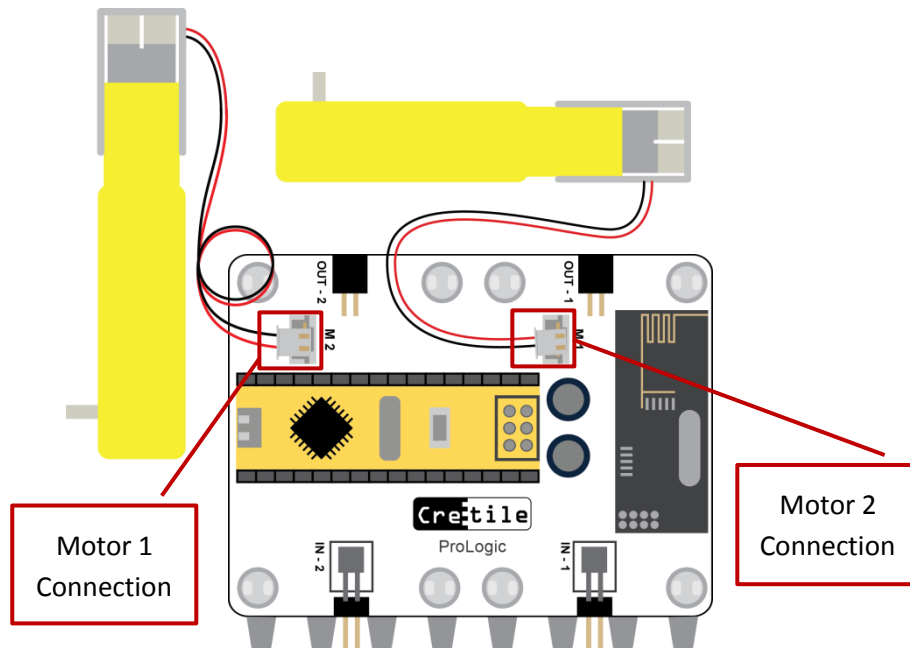
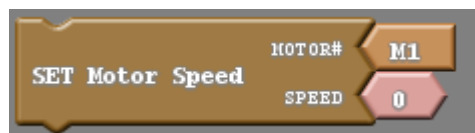


Image 31

There are blocks in ArduBlock to control speed and direction of motors.

a. SET Motor Speed

SET Motor Speed block is used to control speed of MOTOR directly connected to ProLogic. SET MOTOR Speed block appears in ArduBlock as shown below.



b. SET Motor Direction

SET Motor Direction block is used to control direction of motors directly connected to ProLogic. SET MOTOR DIRECTION block appears in ArduBlock as shown below.



7. Using Cretile with Android

Introduction

App Inventor is a tool by MIT which provides Scratch like graphical user interface to develop Android application. Cretile provides extension in App Inventor for communicating with ProLogic using Android phone's USB port. This communication works only when phone has USB-OTG support.

ProLogic local inputs and remote inputs are sent to Android phone, Android phone then can read these values. Android phone can send values of ProLogic local outputs and remote outputs over USB. In ProLogic program you can choose to write values sent by Android phone on respective outputs. There are blocks provided in App Inventor as well as ArduBlock for this communication. In following sections you will learn use of these blocks for communication between Android Phone and ProLogic.

Setting up App Inventor

App Inventor runs in browser. There are few versions of App Inventor available online.

1. <http://ai3.appinventor.mit.edu/>
2. <http://gold.appybuilder.com>

First version is from MIT who have developed App Inventor. Second version is external version with few enhancements. You can use any version according to your preference.

We will use ai2.appinventor.mit.edu.

Installing offline version of App Inventor

You can also install offline version of App Inventor called App Inventor Ultimate from here <https://sourceforge.net/projects/ai2u/>

If you have received USB drive with all Cretile content then you can use the downloaded copy of the installer from your USB drive.

There will be following executable installers in pen drive that you have received.

If you don't have USB drive with content part of your Kit then you can download it from this link <https://sourceforge.net/projects/ai2u/>.

- **AI2U 32bit v4.3.exe** : Use this installer if you are using 32 bit Windows operating system
- **AI2U 64bit v4.3.exe**: Use this installer if you are using 64 bit Windows operating system.

Once you start installer by double clicking on installer executable then you will be asked for permission to install the application and if you allow then you will

see following window. Click on next and follow instruction to install App Inventor offline version.

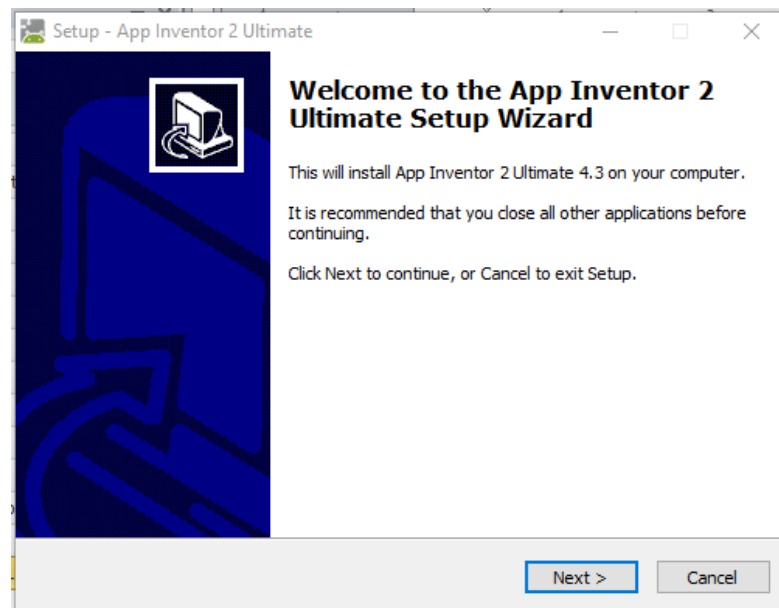
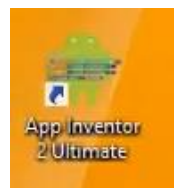


Image 32

When installation finishes you can start the App Inventor server by double clicking on icon for App Inventor 2 Ultimate which will appear as shown below on your desktop or applications.



First time login to App Inventor

Enter url <http://ai2.appinventor.mit.edu/> in chrome browser address bar. If you are using locally installed offline version of App Inventor then you can enter address <http://localhost:8888> in chrome browser address bar.

When you open App Inventor for first time, it asks for a login. You can use your Gmail account to login.

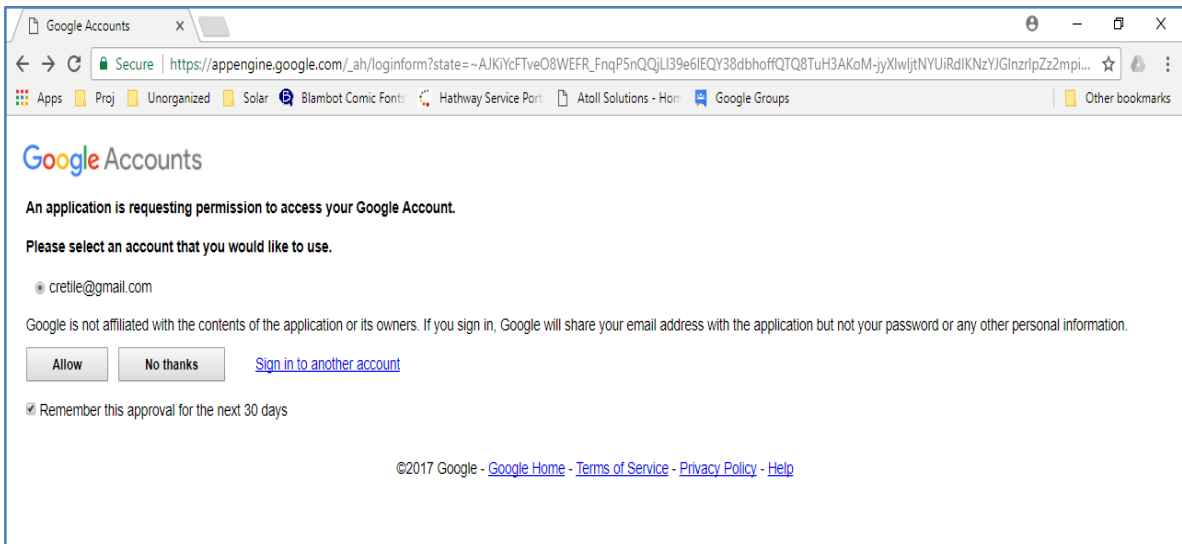


Image 33

You will be presented with terms of service, please read the terms and accept or reject depending on whether you agree to terms or not. If you reject the terms of service then you will not be able to use App Inventor. Rest of the document will be useful to you after you accept the terms of service and proceed to use App Inventor.

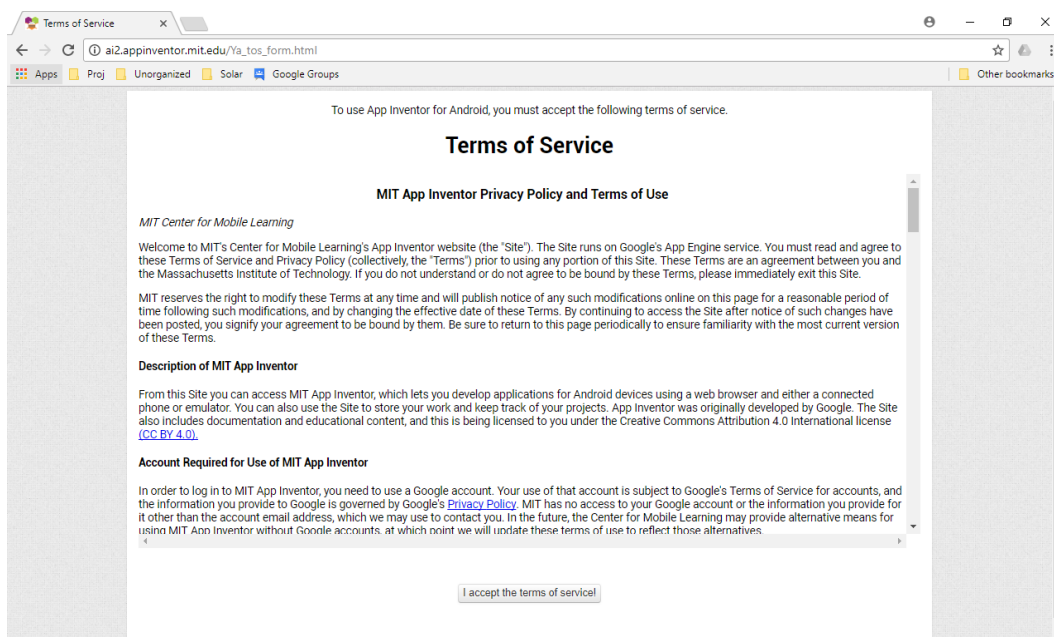


Image 34

Creating project in App Inventor

After login, under Project menu start a new project and give it a name of your choice.

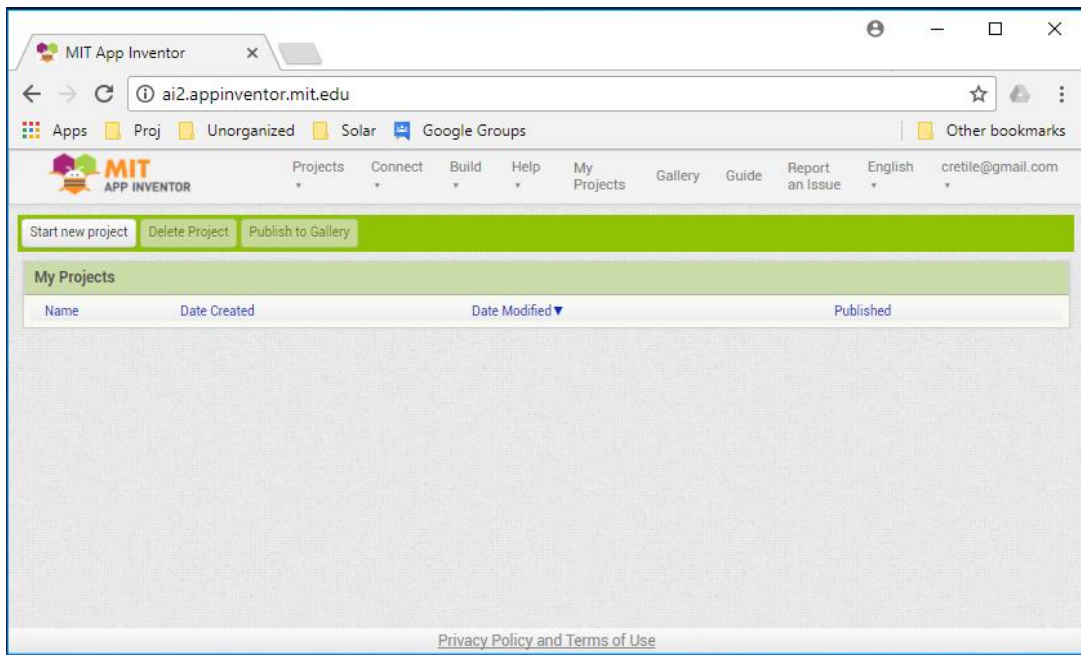


Image 35

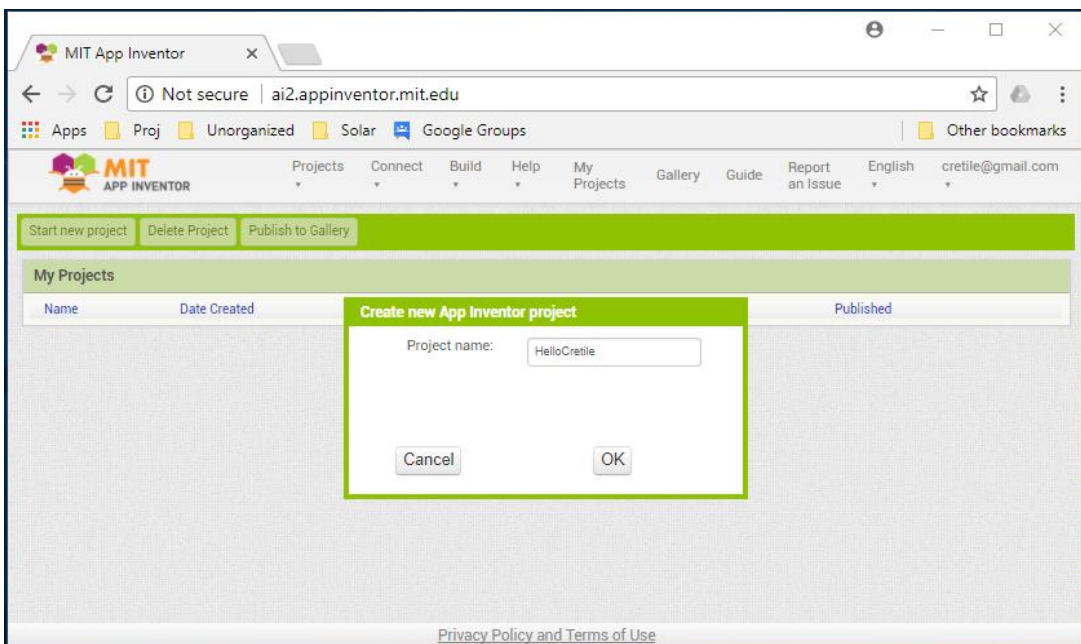


Image 36

Project automatically opens for development after it is created. On left hand side of page you can see many component drawers similar to ArduBlock Block drawers. Each component drawer contains various components which can be used in your application. Some components are GUI components and some are background components which can be used in actions of GUI components e.g. when button is pressed in application. Below figure shows home page of project along with component drawers on left hand side of the page.

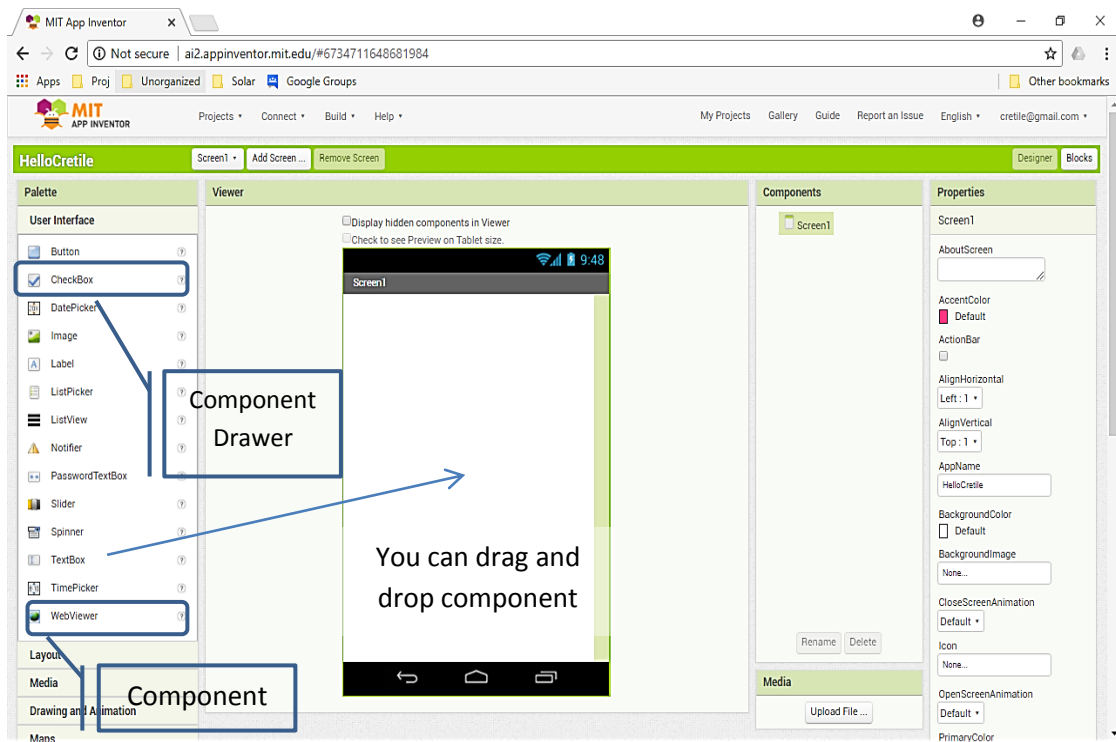


Image 37

Using Android App Inventor extension

App Inventor has way to enhance its capabilities using extensions. A Cretile extension has been created to use ProLogic capabilities in Android applications developed using App Inventor.

After unzipping Cretile.zip you will find folder AppInventorExtension in unzipped files as shown in image. In this folder you will find the App Inventor extension. This is a “ProLogic.aix” file which can be used as extension to App Inventor.

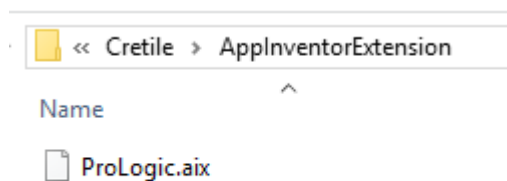
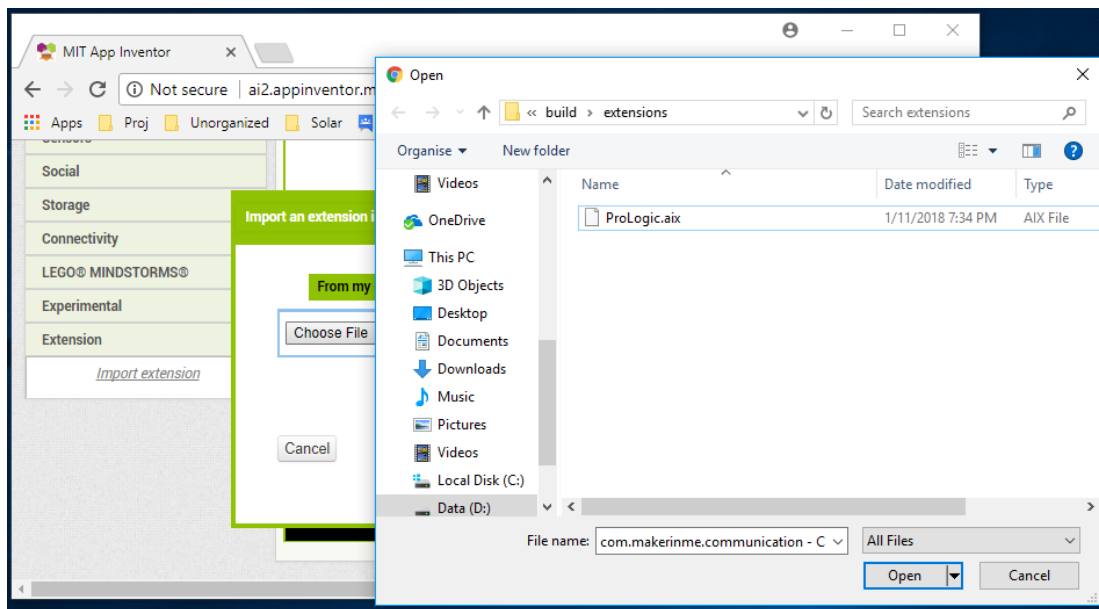
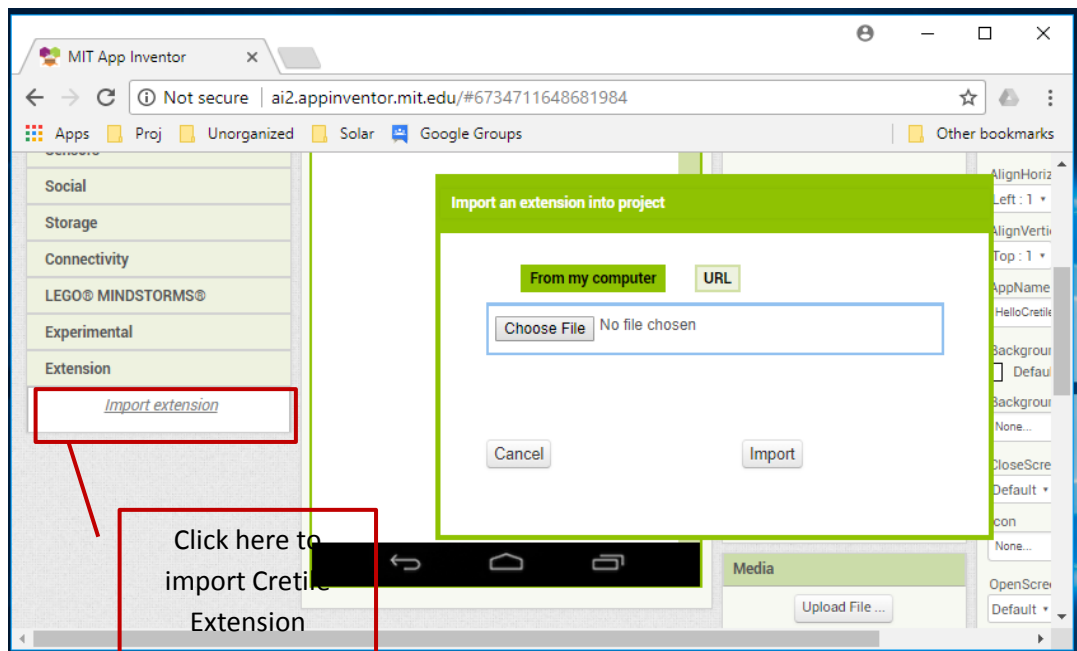
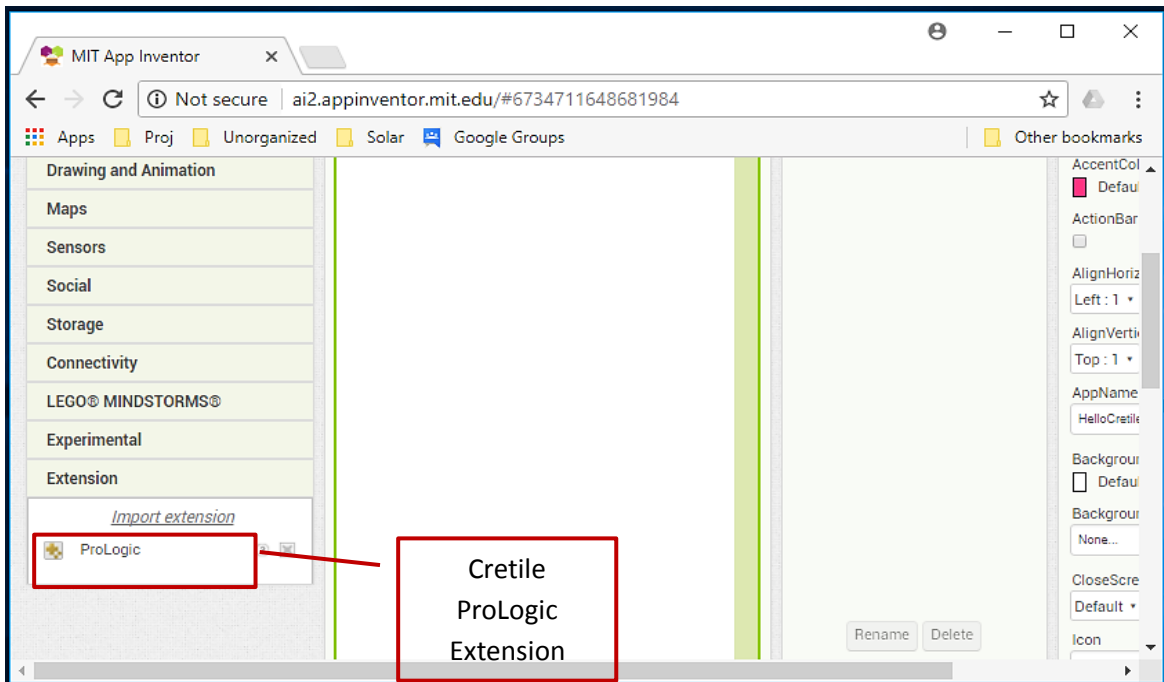


Image 38

Last component drawer in list of component drawers is Extensions component drawer. Click on this component drawer to expand it. Import Extension link will appear. When you click on import extension link a dialog box opens as shown below.



Choose file with name ProLogic.aix as shown in above figure. ProLogic.aix is available in downloaded zip file as shown Image 38 . Click on import button in dialog box to import extension. ProLogic will appear as component in Extensions component drawer as shown below.

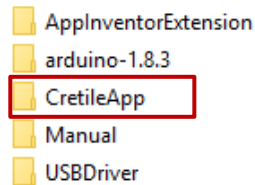


Now you are ready to use ProLogic extension for App Inventor to develop Android-ProLogic applications.

Setting up your Android phone

You need to install Android application provided by Cretile to enable communication between ProLogic and Android application developed by you. You don't have to open the Cretile application as it automatically runs in background when needed by Application developed by you in App Inventor.

When you unzip Cretile.zip you will find CretileApp folder in unzipped folder as highlighted in Figure below.



In this folder there will be Cretile.apk file present which need to be installed on your android phone. To install this apk file you need to transfer this file to your android phone. Please copy this file to Android phone Downloads folder. Please use help on transferring data from PC to Android phone provided on this link [How to Transfer Files From PC to Android Using USB Cable](#). Alternatively, you can search for “How to Transfer Files from PC to Android Using USB Cable” in search engine on internet.

Please enable installation of application from Unknown Sources to install this application on your android phone. You can take help provided on this link [How to Install Apps from Unknown Sources in Android](#). Alternatively, you can search for “How to Install Apps from Unknown Sources in Android” in search engine on internet

Developing Cretile Android App in App Inventor

Cretille App Inventor blocks

We will go through description of all blocks available in App Inventor Cretille extension. Later we will write a simple application which reads local IN-1 of ProLogic and display that on screen of phone. You can skip to section [[Developing android application](#)] and start developing Android Application but understanding all blocks in App Inventor for Cretille will be useful.

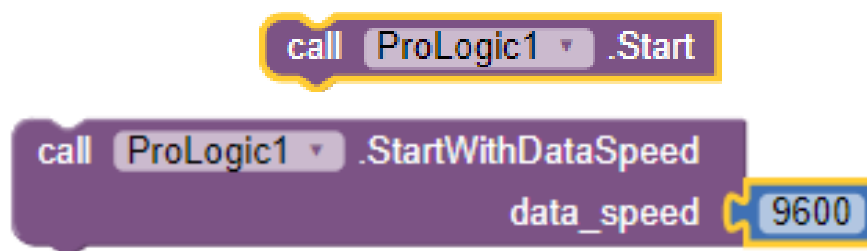
a. *ProLogic.Start*

ProLogic.Start block is used to initialize Cretille android component for communication with ProLogic. There are two types of ProLogic.Start blocks. One is without any arguments and another with an argument of number type.

ProLogic.Start block without arguments have default communication speed between phone and ProLogic set to 115200 bits/second.

ProLogic.Start block with speed argument is used to initialize Cretille android component for communication with ProLogic at given speed. The same speed should be set in ProLogic program with which the phone application will be communicating with.

This block appears in App Inventor as shown below.



b. *ProLogic.ConnectionStatusChanged*

This is a event which occurs when there is any change to connection status between phone and ProLogic e.g. USB cable is plugged or unplugged.

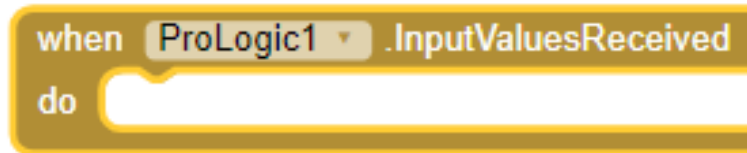
You can add commands in body of this block. All the commands added there will be executed whenever there is change in connection status between ProLogic and Android phone.



c. *ProLogic.InputValuesReceived*

This is an event which occurs whenever new data is received by phone sent by ProLogic.

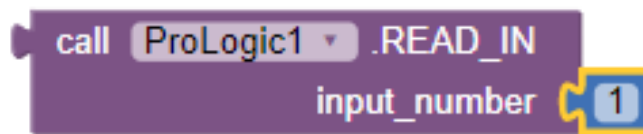
You can add commands in body of this block. All commands added there will be executed whenever new data is received by phone. You can update values of user interface component or perform any action by reading latest data available.



d. *ProLogic.READ_IN*

ProLogic.READ_IN block is used to read local input value sent by ProLogic. This block takes `input_number` as argument and gives value of ProLogic local input at `input_number`. For example, if you provide 1 as argument to this block then it will give value of ProLogic local IN-1 port. *ProLogic.READ_IN* will be able to provide latest value only when ProLogic send the data to phone.

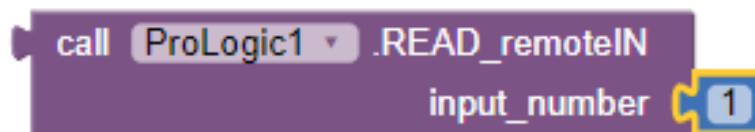
ProLogic.READ_IN block appears in App Inventor as shown below.



e. *ProLogic.READ_remoteIN*

ProLogic.READ_remoteIN block is used to read remote input value sent by ProLogic. This block takes `input_number` as argument and gives value of ProLogic remote input at `input_number`. For example, if you provide 1 as argument to this block then it will give value of ProLogic remoteIN-1 port. *ProLogic.READ_remoteIN* will be able to provide latest value only when ProLogic send the data to phone.

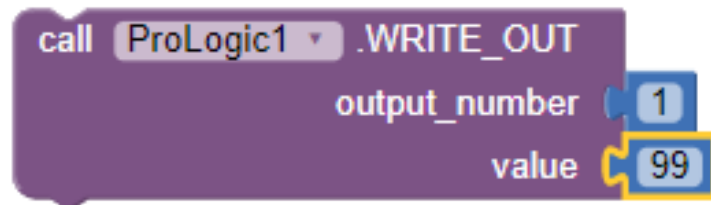
ProLogic.READ_remoteIN block appears in App Inventor as shown below.



f. *ProLogic.WRITE_OUT*

ProLogic.WRITE_OUT block is used to send value for local output of ProLogic. This block takes `output_number` and value to be sent to ProLogic as arguments. For example, if you provide 1 as `output_number` argument and 99 as value argument to this block then it will send value 99 to ProLogic for local output OUT-1. However this value will not appear on OUTPUT until ProLogic program use this value to update OUT-1

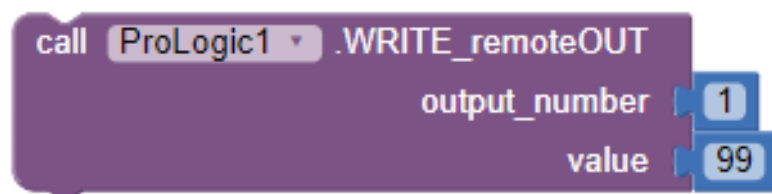
ProLogic.WRITE_OUT block appears in App Inventor as shown below.



g. *ProLogic.WRITE_remoteOUT*

ProLogic.WRITE_remoteOUT block is used to send value for remote output of ProLogic (which appears on Receiver module). This block takes `output_number` and value to be sent to ProLogic as arguments. For example, if you provide 1 as `output_number` argument and 99 as value argument to this block then it will send value 99 to ProLogic for local output OUT-1. However this value will not appear on remote OUTPUT until ProLogic program use this value to update remote OUT-1.

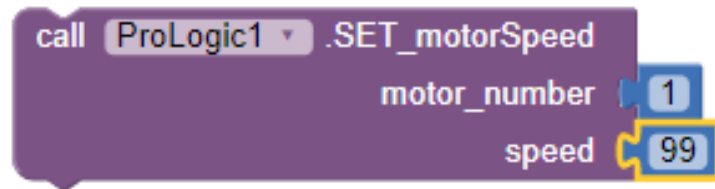
ProLogic.WRITE_remoteOUT block appears in App Inventor as shown below.



h. *ProLogic.SET_motorSpeed*

ProLogic.SET_motorSpeed block is used to send value for motor speed for motors connected to on-board motor drivers. This block takes `motor_number` and speed value to be sent to ProLogic as arguments. For example, if you provide 1 as `motor_number` argument and 99 as speed argument to this block then it will send value 99 to ProLogic for motor M1. However speed of motor M1 will not be updated until ProLogic program use this value to update motor speed.

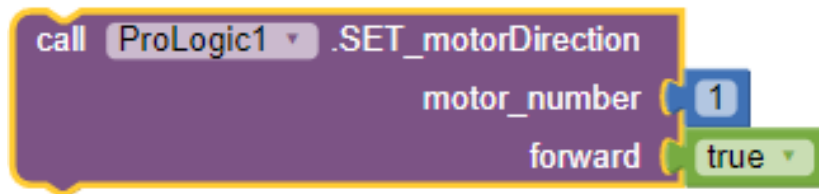
ProLogic.SET_motorSpeed block appears in App Inventor as shown below.



i. *ProLogic.SET_motorDirection*

ProLogic.SET_motorDirection block is used to send value of motor direction for motors connected to on-board motor drivers. This block takes motor_number and forward value to be sent to ProLogic as arguments. For example, if you provide 1 as motor_number argument and forward is set to true then it will send direction forward to ProLogic for motor M1. However direction of motor M1 will not change until ProLogic program uses this value to update motor direction.

ProLogic.SET_motorDirection block appears in App Inventor as shown below.



Developing android application

To display value you will need Label component in the application. Drag and drop Label component 2 times from User Interface component drawer to Screen of application as shown below. Rename the labels to textIN1 and valueIN1. “textIN1” label will always display ProLogic.IN1. You can set initial text value in label “Text” property of Label as shown below.

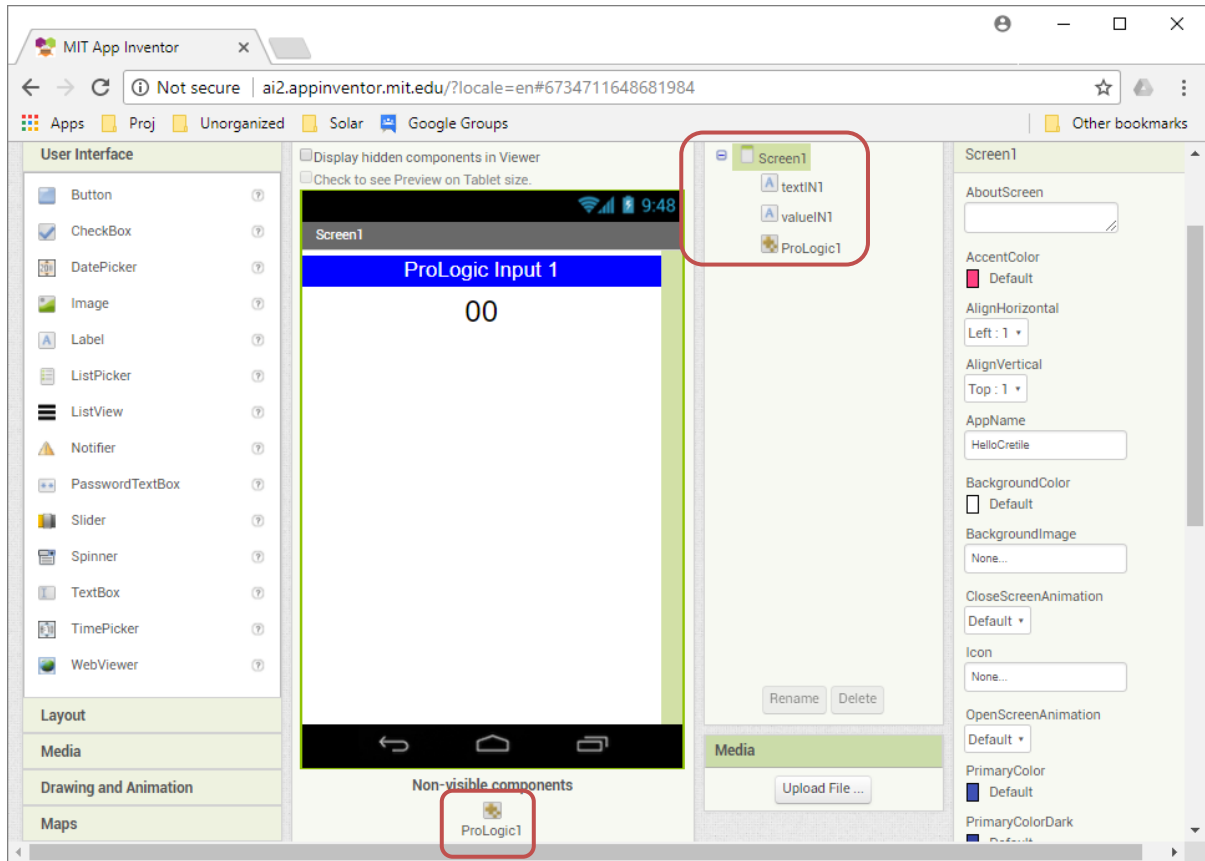


Image 39

Drag ProLogic component from Extension component drawer and drop it to Screen1. After this component instance ProLogic1 will appear under Screen1 as shown in above image.

To communicate with ProLogic, first ProLogic component on Android phone need to be made ready for communication. Click on “Blocks” button to switch to blocks mode where various possible actions are shown for each component added in the application. ProLogic component will be visible as shown in below figure.

Click on Screen1 and drag&drop **When Screen1.Initialize** block. This block is executed when Application starts.

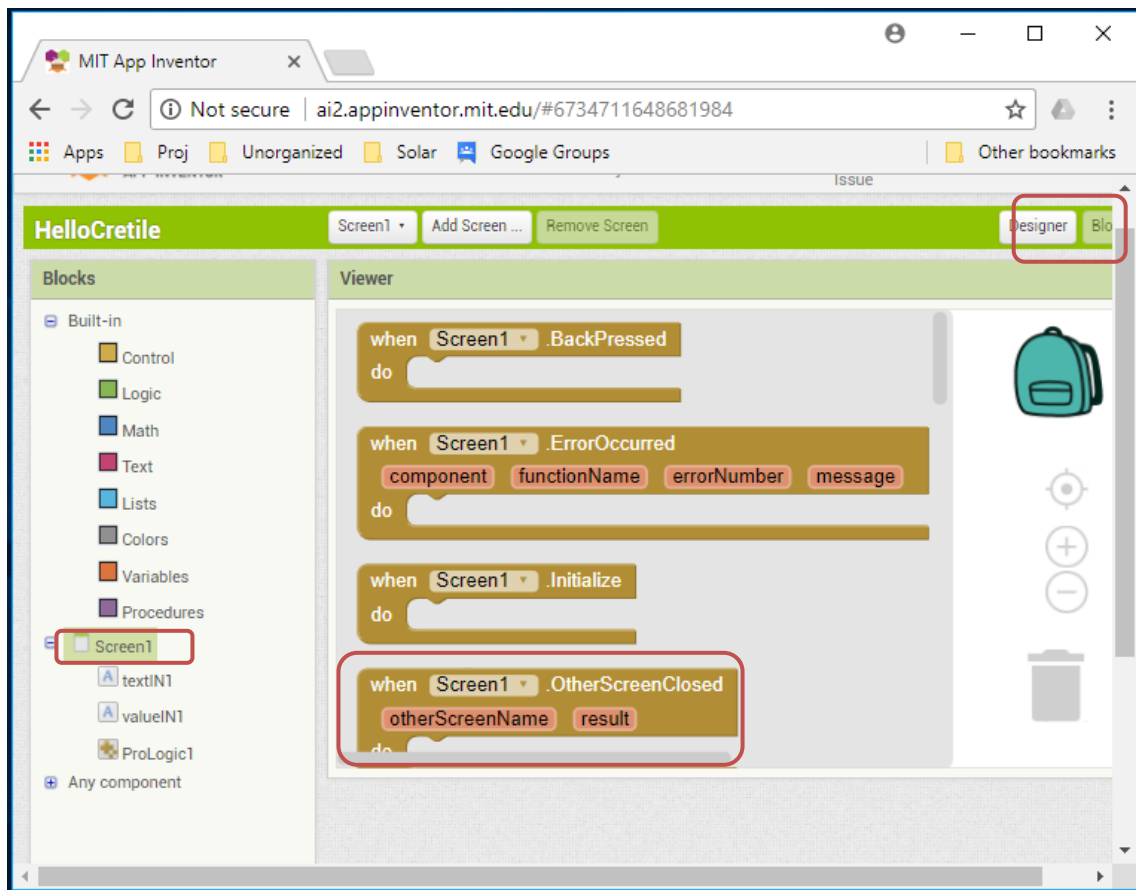


Image 40

Click on ProLogic component and all blocks related to ProLogic component will be visible as shown below Figure.

Select **ProLogic1.Start** block and add it in event **Screen1.Initialize**. This will initialize ProLogic Android communication component with default data speed of 115200 bits per second.

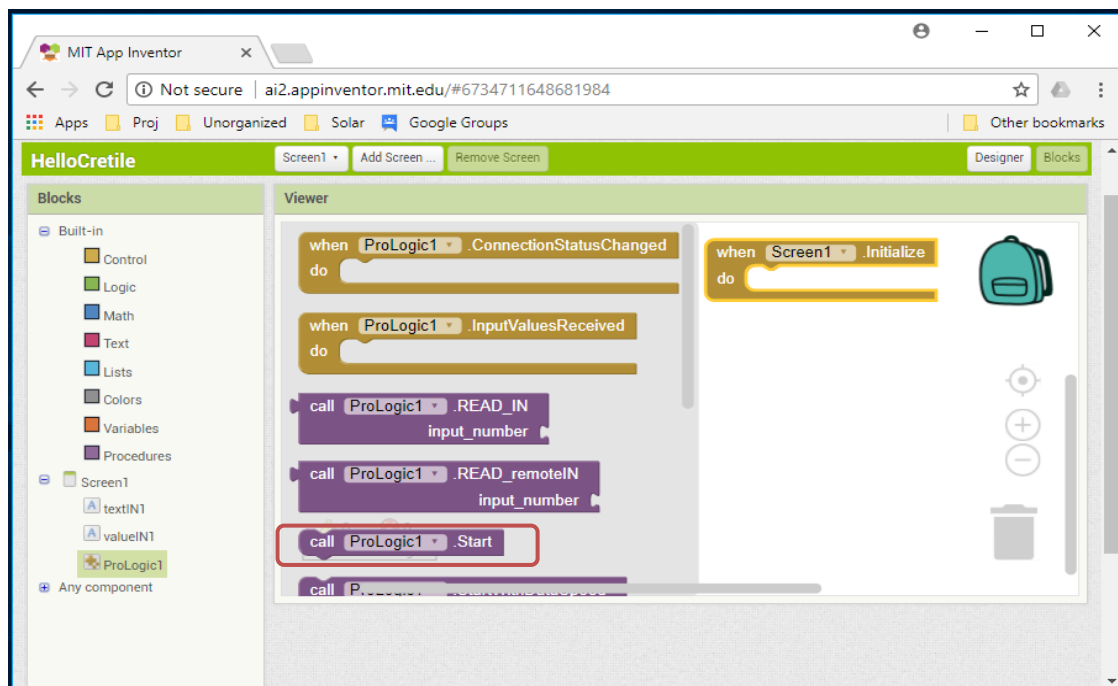


Image 41

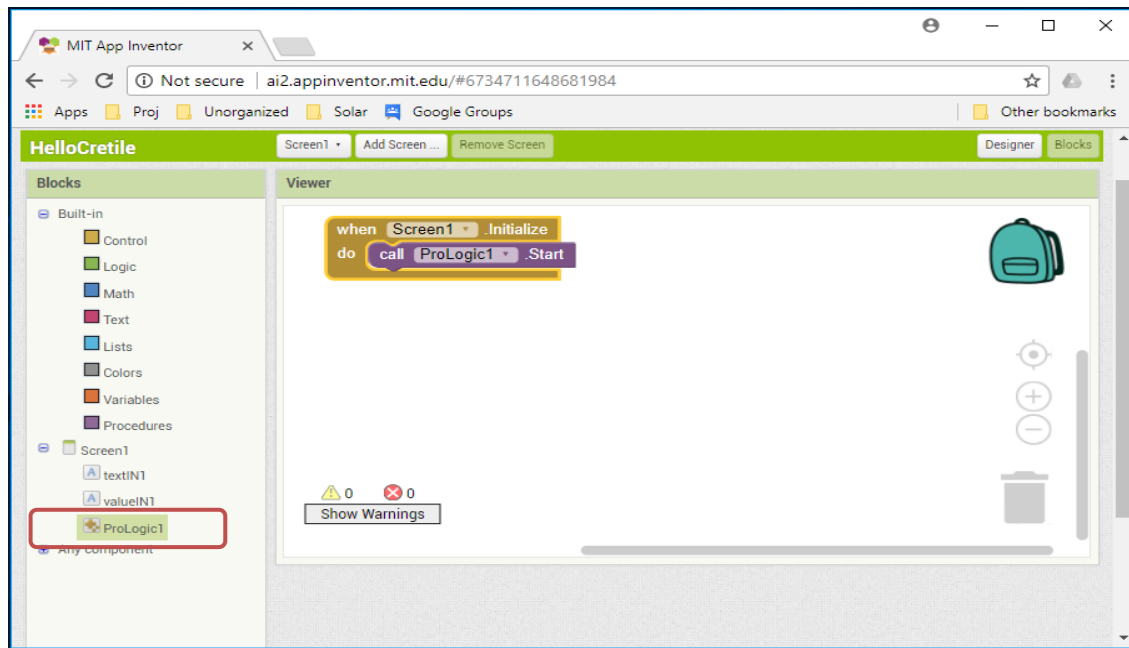


Image 42

From ProLogic1 component under Screen1, drag&drop **when ProLogic1 InputValuesReceived** block to canvas. This block is executed whenever ProLogic input values are received by Android phone.

Now Click on valueIN1 component under Screen1 and select **set Text to** block and drag& drop this block in **when ProLogic1 InputValuesReceived** as shown in Image 44.

Click on ProLogic1 component. Drag& drop **ProLogic.READ_IN** block to attach the block to **set Text to** block of valueIN1 as shown in below Figure Image 44.

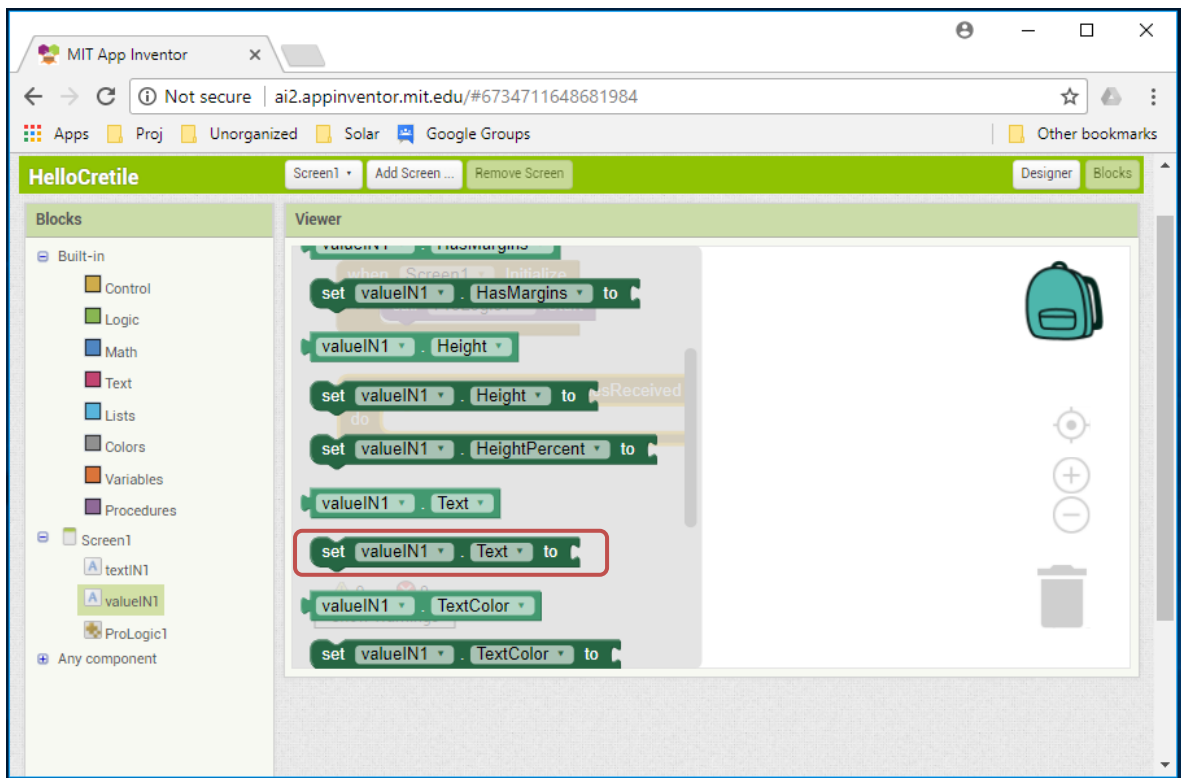


Image 43

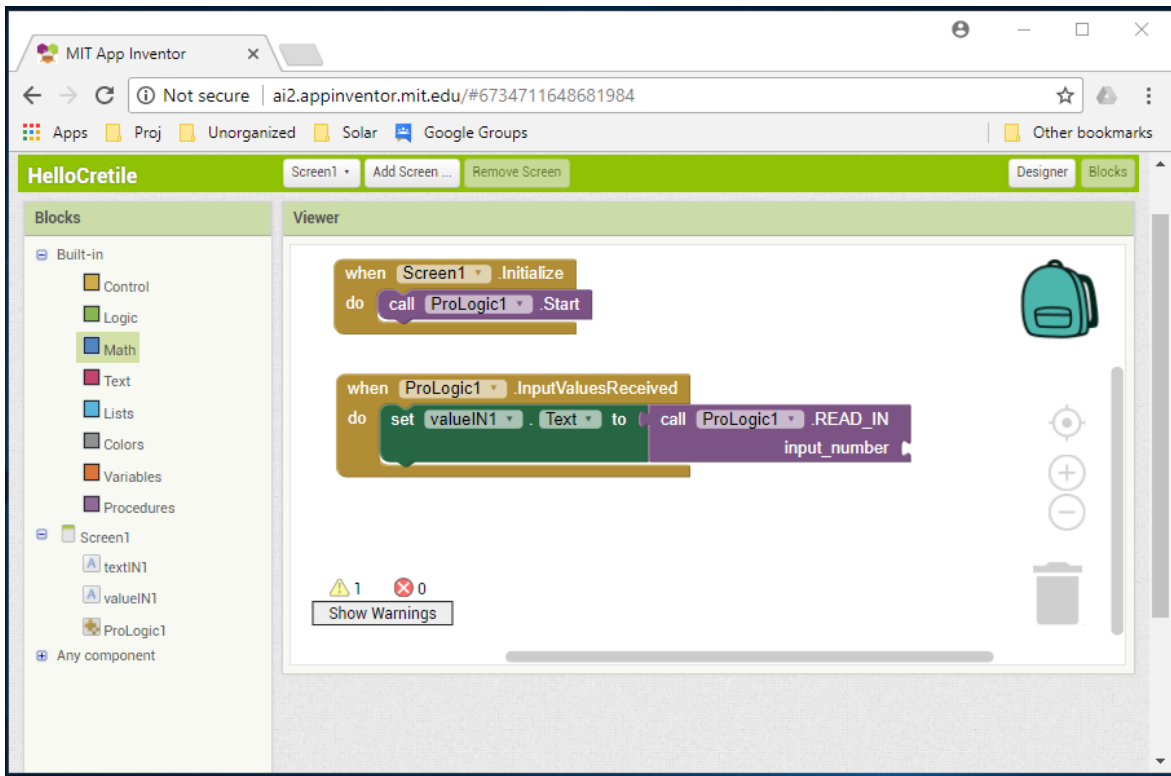


Image 44

Drag & drop number block from Math component drawer and attach it to **ProLogic1.READ_IN** block and set number to 1 as shown in two figures below. This will read value of ProLogic local input **IN-1**. Text of valueIN1 will get updated with recent value of ProLogic **IN-1**.

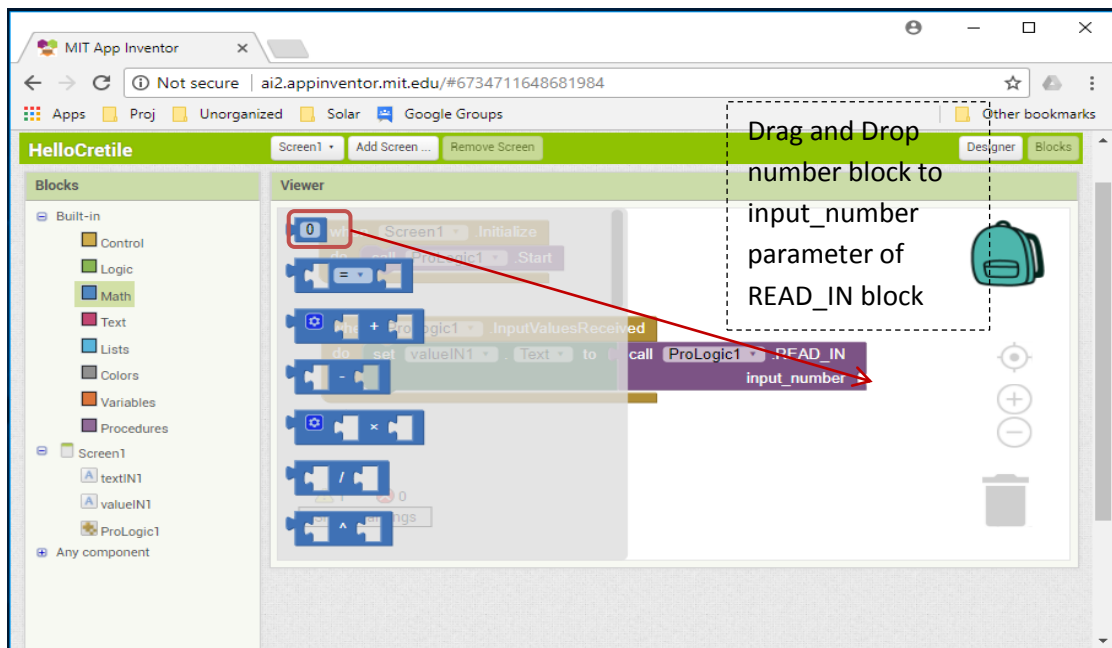


Image 45

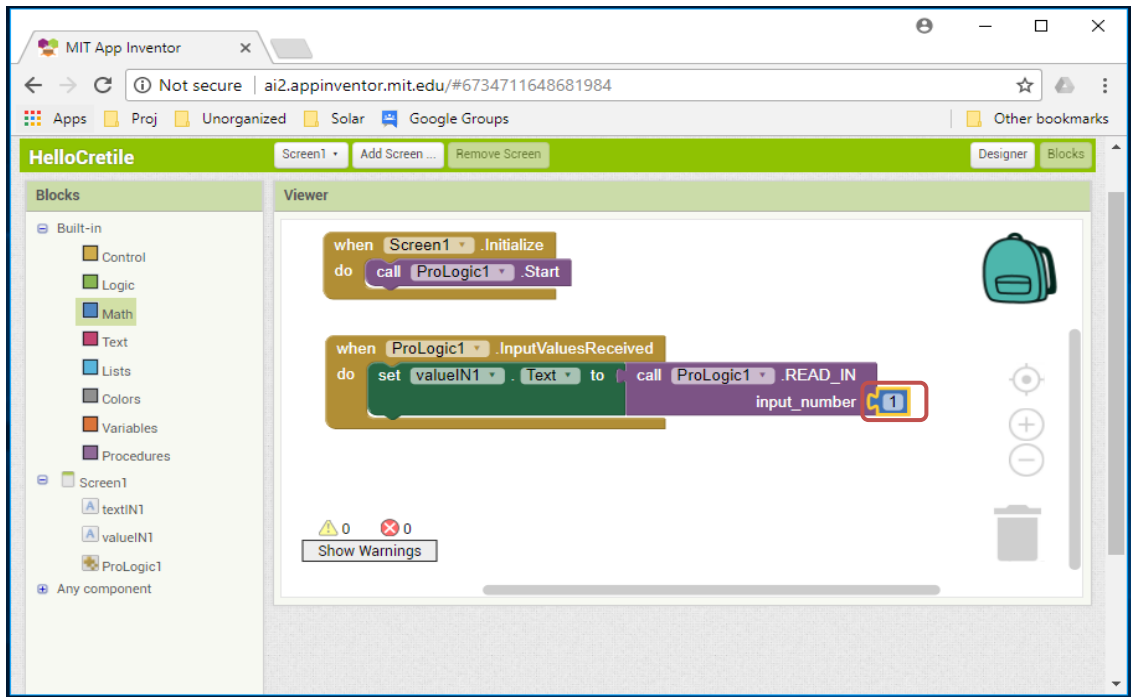
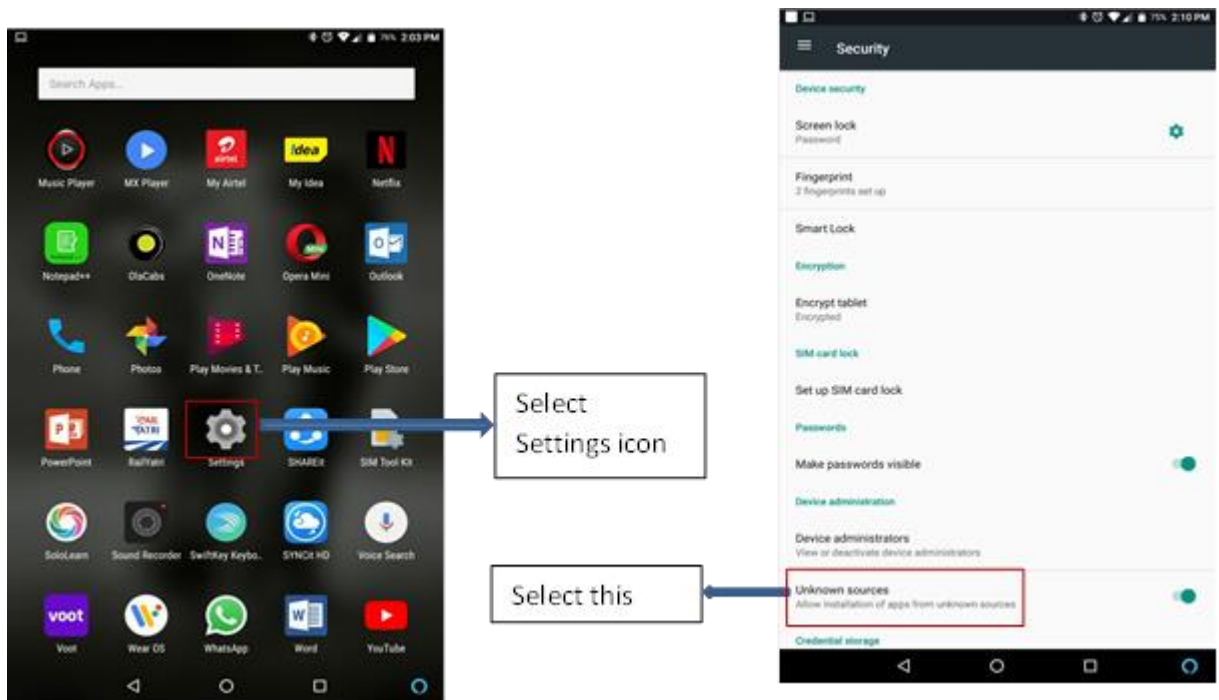


Image 46

Installing application on android phone

To install application developed in App Inventor, You should Allow installation of Apps from Unknown sources in Android Phone by following the below steps



Go to main menu in Android Phone, Select Settings and select security as shown below

Image 47

Select Unknown source and select OK for warning message.

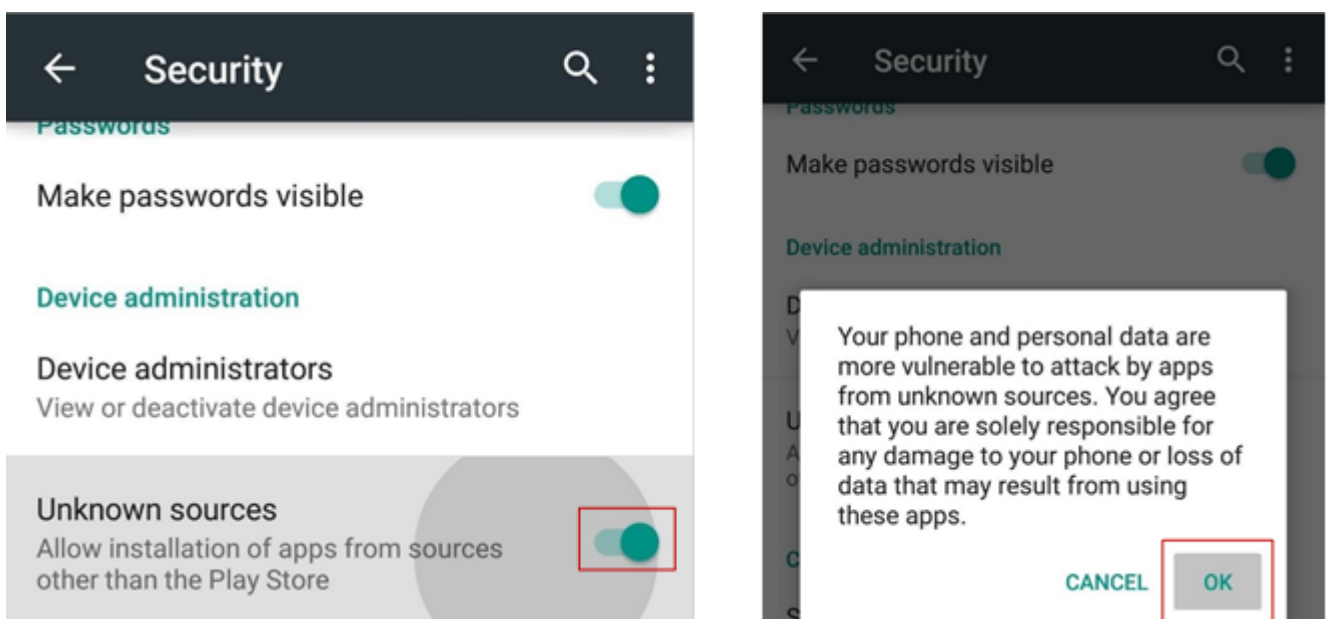
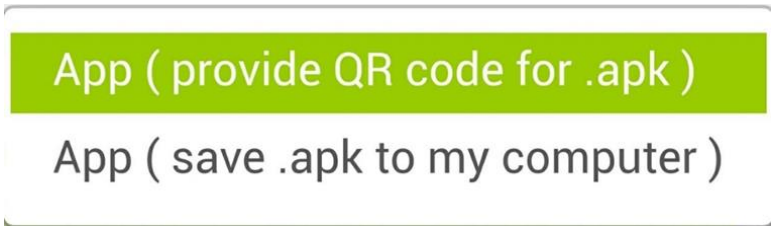
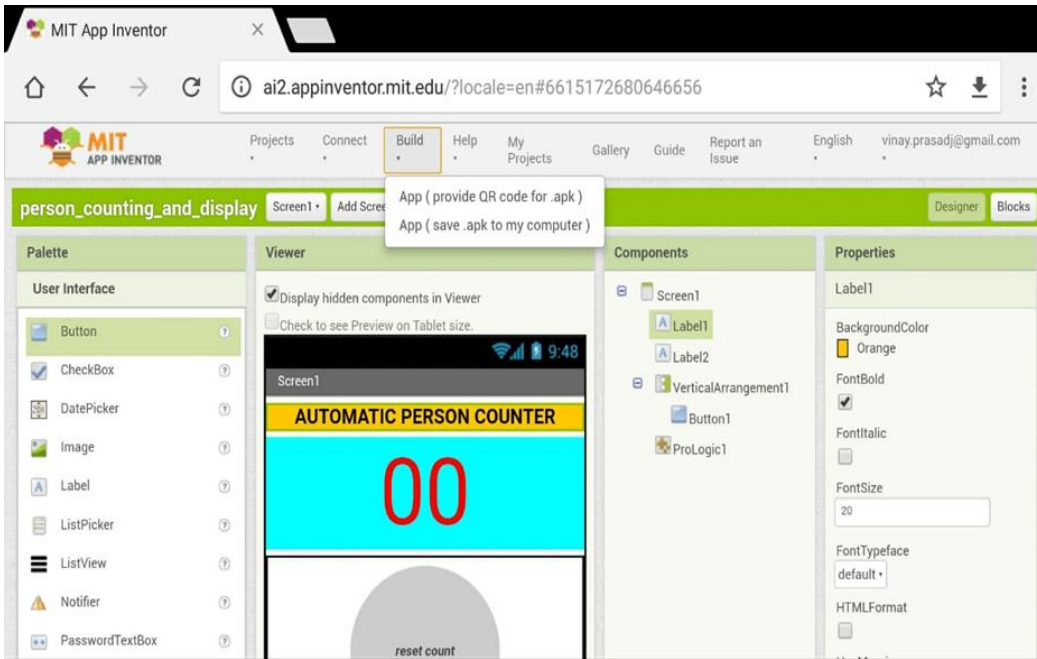


Image 48

Select build option at the top of MIT App Inventor window and
Select **“App (provide QR code for .apk)”**, as shown in below diagrams



Now you can see window below, which indicate progress of compiling

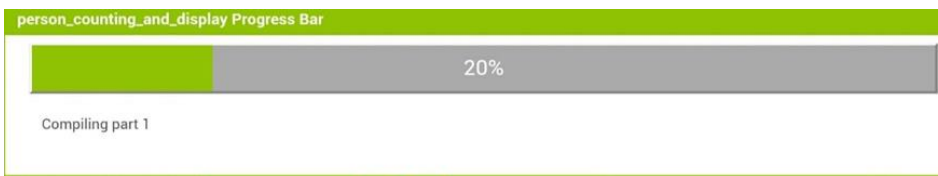
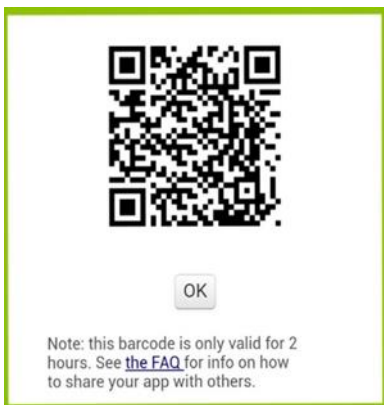


Image 49

We can see the QR code here as below



Open MIT AI2 Companion application in your Android phone and Scan QR code shown in below image.

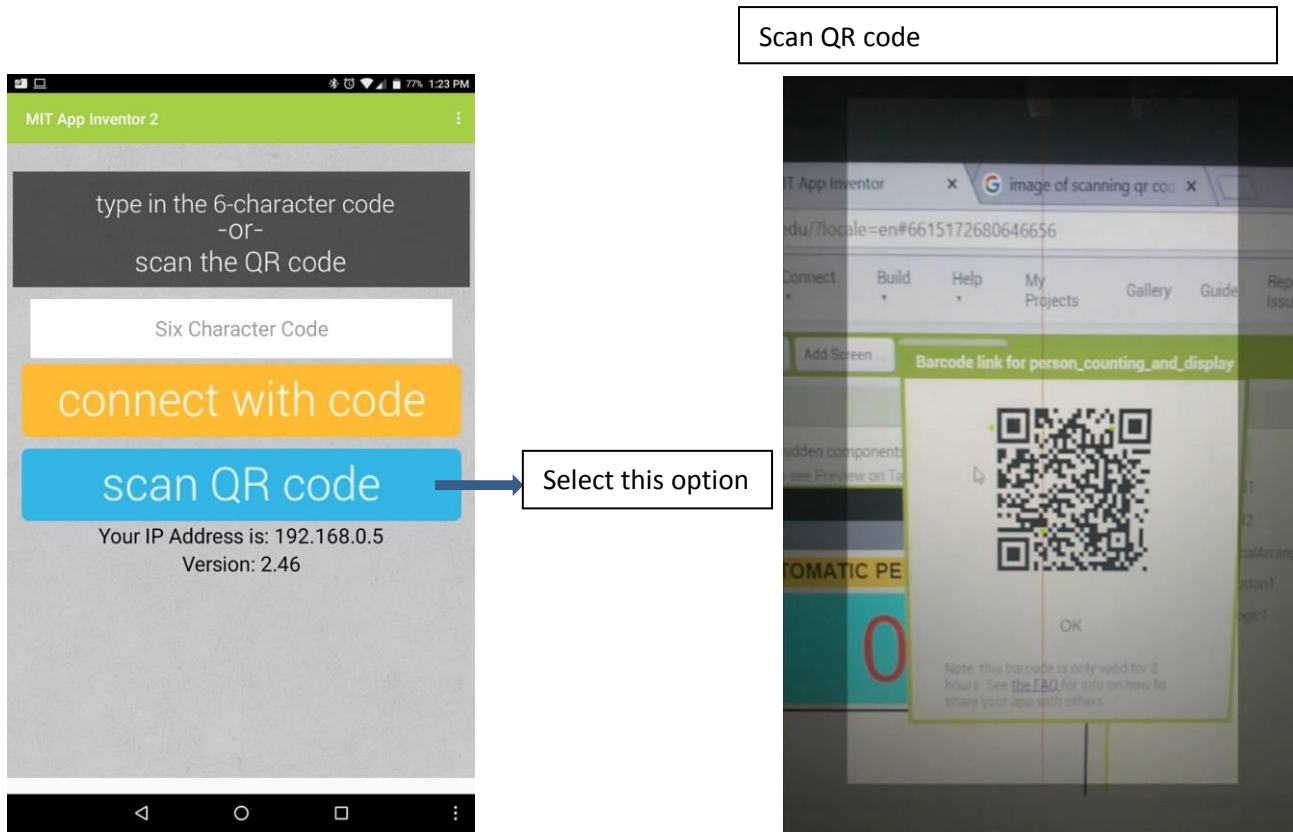


Image 50

Once the QR code is successfully scanned, your app will get installed on Android phone.

Developing ProLogic program to work with the Android application

Blocks for Mobile/PC communication in ArduBlock

ArduBlock for Cretille has block drawer named *Cretille ProLogic phone/PC*. This block drawer contains blocks to send data to phone/PC and use data from Android phone/PC.

Open Arduino->ArduBlock for Cretille as explained in section [[Setting up software for programming](#)]. There is Block drawer named *Cretille ProLogic Phone/PC* which provides all the blocks needed for communication between phone/PC and ProLogic (using USB cable).

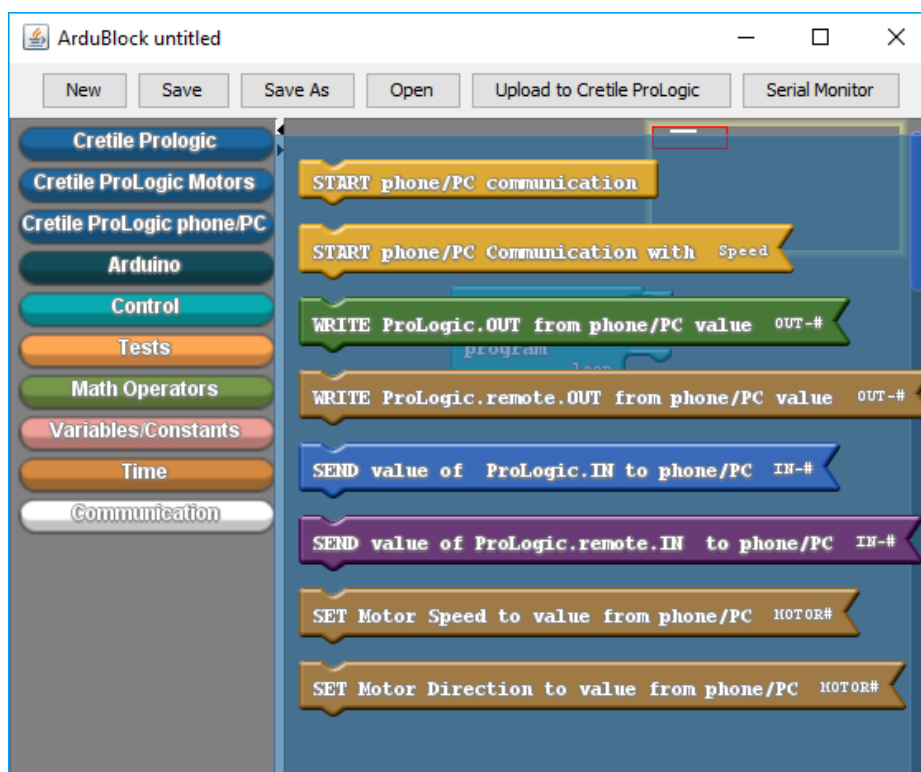
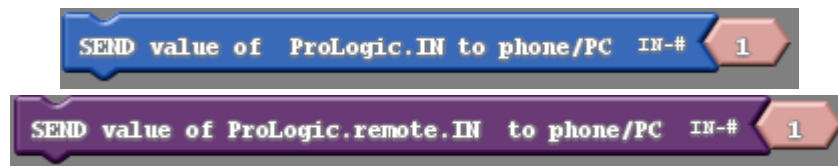


Image 51

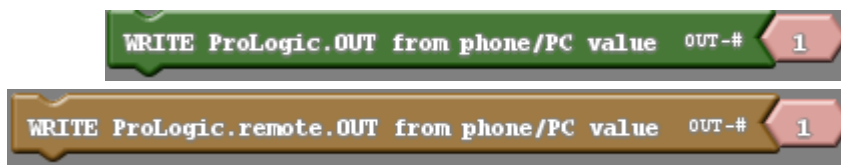
To start communication with smartphone/PC you need to initialize ProLogic. For this *Start Phone/PC communication* block is used. Start phone/PC communication block appears in ArduBlock as shown below.



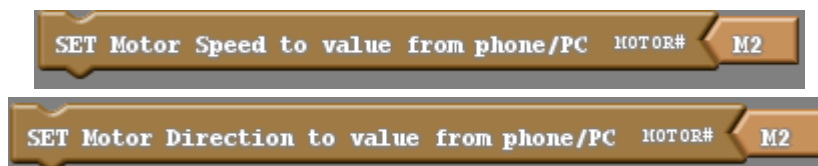
You can send values of ProLogic local and remote INputs to smartphone. For this *SEND value of ProLogic.IN to phone/PC* and *SEND value of ProLogic.remote.IN to phone/PC* blocks are used. You need to specify INput number for which you are sending values. These blocks appear in ArduBlock as shown below.



Even though ProLogic receives data from phone/PC for values of OUTputs automatically always, those are not directly written on OUTputs. You can choose to write values of local and remote OUT whenever you want in program. You can choose to update one or more OUTputs when required. This can be done by **WRITE ProLogic.OUT from phone/PC** value and **WRITE ProLogic.remote.OUT** from phone/PC blocks. These blocks need output number as parameter. These blocks appear in ArduBlock as shown below.



Similar to local and remote OUTputs, on-board Motor drivers in ProLogic can be controlled from Android application. Similar to local and remote OUTput values Motor speed and direction values are received by ProLogic when you use **Start phone/PC communication** block. You can choose to update motor speeds and direction using the received values whenever you want in your program. This is achieved by using **SET Motor speed to value from phone/PC** and **SET Motor direction to value from phone/PC**.



Developing ProLogic Program

Android Application developed in section 3.3.1 need value of IN-1 to be sent from ProLogic to the phone. You need to develop and upload program on ProLogic which will do this job.

ProLogic program should do following

1. Initialize Phone/PC communication in setup section
2. SEND value of local IN-1 to Mobile/PC in loop section
3. Wait for 10 milliseconds – by doing this you are sending IN-1 value every 10 millisecond i.e. 100 times in a second.

Complete program is shown below.

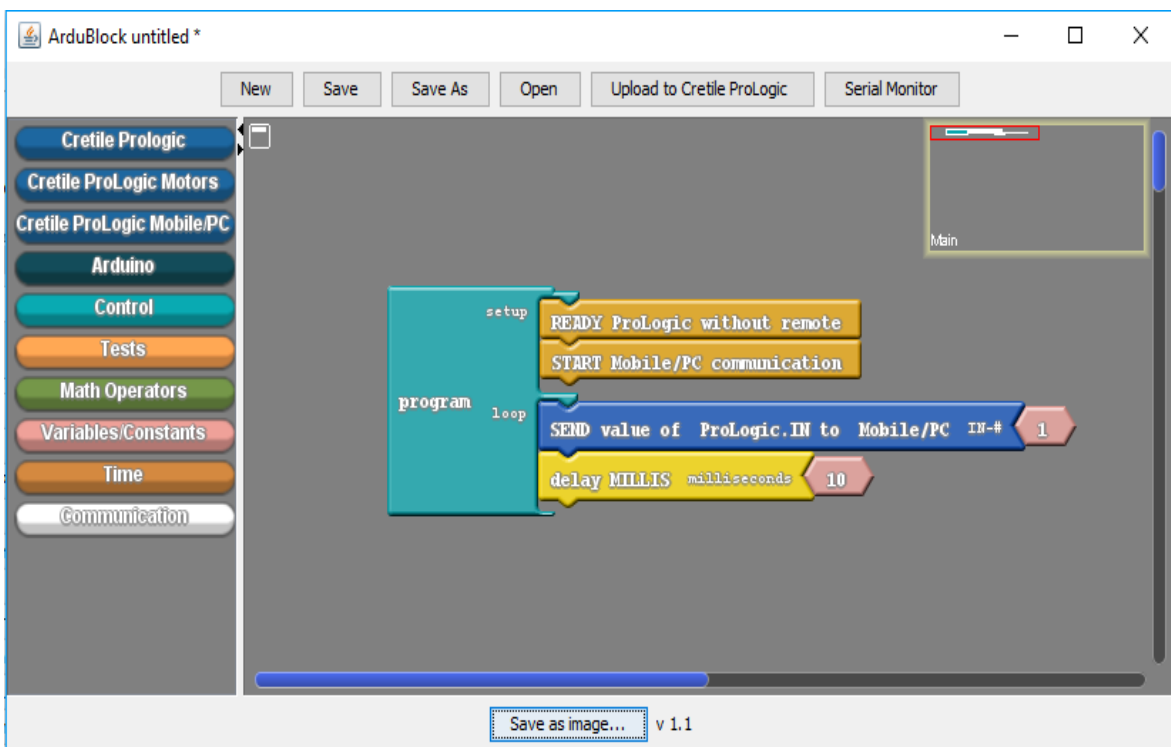


Image 52

Upload this program to ProLogic using Upload to Cretile ProLogic button as mentioned in Section [

Introduction to ArduBlock].

Connecting Android phone with ProLogic using USB-OTG

Start application created in section [Developing android application] in android phone and connect the phone with ProLogic which is programmed in section [

Developing ProLogic Program]

You will need USB-OTG adapter/cable to connect ProLogic to your USB-OTG capable android phone. The setup is show below in Figure 20.

Once phone is connected to ProLogic it will be recognized by the App and permission request for accessing USB device will appear. Grant the permission and communication with ProLogic will be established. In this setup we have connected dimmer on IN-1 of ProLogic so application will display value of dimmer. If you change value of dimmer, updated value will be shown in the application.

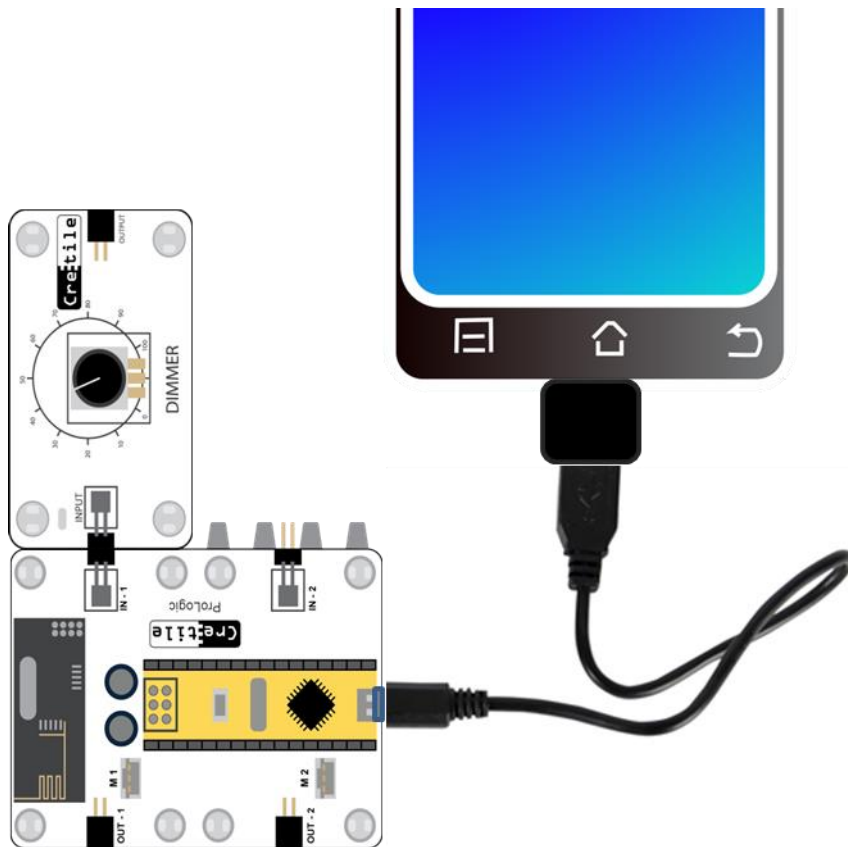


Image 53

Thank you for purchasing CreTile. Please read the instruction manual carefully before using the product. Please ensure that the product is used under adult supervision.

BASIC INSTRUCTIONS

- Always assemble or de-assemble CreTiles in power-off mode

- Always ensure that the product is used under adult supervision and not left unattended.
- Always ensure that the circuit does not come in contact with any metal objects or liquids.
- Always ensure that the product is utilized along with a high quality ISI/ISO certified 9V 2A adapter with a compatible DC PIN for the power jack in Power module.
- Always ensure that the product is utilized with two 9V PP3 battery only when using Battery module.
- Always ensure that the battery is connected with correct electrical polarity. Caution should be taken not to short-circuit battery terminals to each other which can damage the product and generate heat that may result in burns.
- Always take maximum precaution while plugging in batteries so that they are not connection to any wrong electrical polarity. Wrong electrical polarity can cause fire or burns.
- Always ensure that the product is used within 0 (zero) to 50 (fifty) degree temperature range only. Exposure to direct sunlight should be avoided.
- Never recharge a non-rechargeable battery.

IMPORTANT: PLEASE READ CAREFULLY

Please ensure that you have read and complied with all the details provided on the brochure/instruction manual placed. Please ensure safe and supervised usage of the entire product. In the event that any of the components of the product are dropped, the product may not function as desired. We do not make any express or implied warranties of any kind with respect to the product and we expressly disclaim any warranties of merchantability or fitness for a particular purpose. By using this product, you acknowledge and understand that the battery and the adapter provided separately have not been manufactured by us and we shall in no event be held liable for any loss or damage that may arise as a result of the use of such battery and the adapter. Kindly ensure that you have read and complied with all the instructions provided for the use of the battery and the adapter by the manufacturer. We shall not be held responsible for any incidental, indirect, consequential, or special damages that may arise as a result of the use of the product. Your use of the product implies your acceptance of these terms. If there are any defects or defaults in the product, kindly contact support@cretile.com within 10 (ten) days from the date of purchase of the product and the same would be replaced.

Any dispute which may arise as a result of the use of this product shall be subject to the jurisdiction of the Courts situated in Bangalore, Karnataka.

Kindly keep the instruction manual and package for reference as they contain important information.

A Product of



www.MakerInMe.com

Copyright @ MakerInMe Technologies Private Limited



Remote Controlled Curtain



Cre-tile
Projector



Bubble
Machine



Door Alarm



Morning Plant Watering System



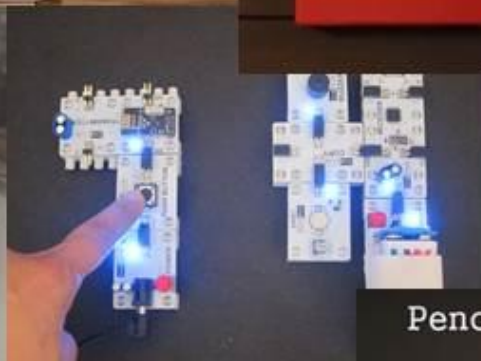
Mail Indicator



Mail Indicator



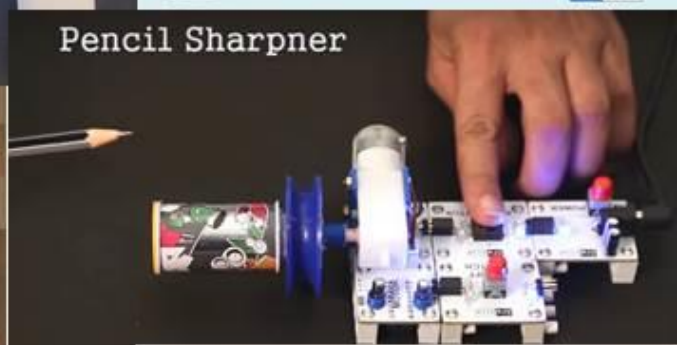
Beep Player



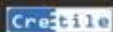
Water Dispenser



Remote Controlled Vehicle



Pencil Sharpener



"Logical NOT" in Physical World



Easy Drag-n-Drop
Programming Environment

Use Programming Language	Use Hands & Visuals
<pre>#include <ard.h> int main(void) { bool inValue = 0; bool outValue = 0; outValue = !(inValue); printf("outValue is %d\n", outValue); }</pre>	<p>Switch is OFF → Light is ON</p> <p>Switch is ON → Light is OFF</p> <p>Play → Observe → Learn</p>



www.CreTile.com