Sensorian.

Programming the Sensorian Shield with Scratch

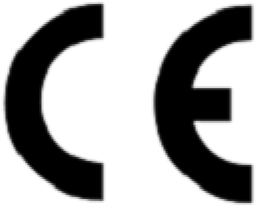
The latest version of this booklet is available online at: www.sensorian.io

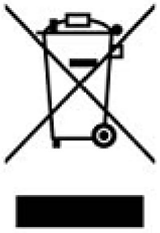
Copyrights

© Sensorian, 2015. The information contained herein is subject to change without notice. Sensorian assumes no responsibility for the use of any circuitry other than circuitry embodied in a Sensorian product. Nor does it convey or imply any license under patent or other rights. The Sensorian shield is not warranted nor intended to be used for medical, life support, lifesaving, critical control or safety applications, unless pursuant to an express written agreement with Sensorian. Sensorian reserves the right to make changes without further notice to the materials described herein. Sensorian does not assume any liability arising out of the application or use of any product or circuit described herein. Sensorian does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Sensorian product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Sensorian against all charges.

Regulatory Compliance

The Sensorian Kit is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. Due to this reason, the board may cause interference to other electrical or electronic devices in close proximity. In a domestic environment, this product may cause radio interference. In such cases, the user may be required to take adequate preventive measures. Also, this board should not be used near any medical equipment or RF devices. Attaching additional wiring to this product or modifying the product operation from the factory default may affect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken. The Sensorian Kit as shipped from the factory has been verified to meet with requirements of CE as a Class A product.





Handling Boards

Sensorian boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static free surface. Use a conductive foam pad if available. Do not slide board over any surface.

Table of Contents

Acknowledgment

Special thanks to Michael Lescisin for his contributions to the development of this booklet.

Introduction

Scratch is a development environment aimed at people learning how to program. Users of the Sensorian Shield who do not yet have a solid understanding of Python or C can still develop applications for Sensorian thanks to our Sensorian API for Scratch. Our API uses the MESH (multi-player) functionality of Scratch to communicate with a Python based server which constantly feeds Scratch with sensor data as well as receives commands from Scratch which manipulate Sensorian outputs. MESH functionality comes loaded, but disabled, by default in Scratch. Our custom image comes with MESH enabled in Scratch.

The default configuration is to have Scratch listening for inbound connections and have our Python server make an outbound connection to Scratch. The Python server is designed to connect, by default, to 127.0.0.1:42001 which is the Scratch instance running locally on the Raspberry Pi. The Python server must be started with the list of Sensorian sensors which the user wishes to share with Scratch. Optionally, the server can be instructed to connect to a Scratch instance running on another computer. This is useful for a complex Scratch program running on a desktop computer which uses inputs and outputs from the Sensorian.

To use Sensorian in Scratch on the Raspberry Pi, open Scratch and issue Shift+Click on Share and select “Host Mesh”. Next, open a terminal and run:

```
sudo python ~/Sensorian/Handler_Scratch/run_server.py
```

This will print a help page with a list of arguments. Now, you can start the server with the list of sensors to export to Scratch. For example:

```
sudo python ~/Sensorian/Handler_Scratch/run_server.py --touchpad
```

will export the capacitive touch sensors to Scratch running locally on the Raspberry Pi.

The sample applications can be found under `~/Sensorian/Apps_Scratch`

To use Sensorian in Scratch on a Windows/Mac/Linux computer, open Scratch, issue Shift+Click on Share and select “Host Mesh”. Next, open a terminal on the Raspberry Pi and follow the above steps for running the server and append `--connect IPADDRESS` (IPADDRESS is the address of the Windows/Mac/Linux computer where Scratch is running).

Project 1: Blinking an LED

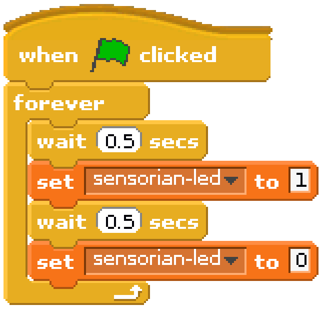
Project idea: The on-board LED can be accessed through the Scratch variable "sensorian-led". Setting it to **one** turns the LED on while setting it to **zero** turns the LED off.

To get started:

1) Create the "sensorian-led" variable. Click on "Variables", "Make a variable", and name the variable "sensorian-led".
2) Create the main program template. Click on "Control" and add the "when green flag clicked" to the scripts window. Next, drag the "forever" block onto the scripts window and attach it to the "when green flag clicked" block in the scripts window.
3) Set the blinking rate of the LED. Add the "wait" block to the inside of the "forever" block.
4) Turn the LED on. Click on "variables" and add the "set" block to the inside of the "forever" block immediately below the "wait" block. Set the variable to "sensorian-led" and the value to **1**.
5) Give the LED time to remain on. Add another "wait" block immediately below the "set" block from the previous step.
6) Turn the LED off. Add another "set" block immediately below the "wait" block from the previous step. Set its variable to "sensorian-led" and the value to **0**.

Final program:

The program should now resemble the following. Clicking the green flag should cause the Sensorian LED to start blinking.



```
when 🚩 clicked
forever
    wait 0.5 secs
    set sensorian-led ▾ to 1
    wait 0.5 secs
    set sensorian-led ▾ to 0
```

Figure 1: The final program. Blinks the Sensorian LED

Next steps:

Try modifying the "wait" blocks to control how the LED blinks.

Project 2: Using the TFT-LCD

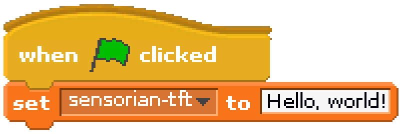
Project idea: The TFT-LCD display will display any text which is contained in the Scratch variable "sensorian-tft"

To get started:

1) Create the "sensorian-tft" variable. Click on "Variables", "Make a variable", and name the variable "sensorian-tft".
2) Create the main program template. Click on "Control" and add the "when green flag clicked" to the scripts window.
3) Write something to the display. Click on "Variables" and add the "set" block to the scripts window immediately below the "when green flag clicked" block. Set the variable to "sensorian-tft" and the value to "Hello, world!"

Final program:

The program should now resemble the following. The LCD should display "Hello, world!" when the green flag is clicked.



```
when ⚑ clicked
set sensorian-tft ▾ to Hello, world!
```

Figure 2: The final program which simply prints "Hello, world!" on the TFT LCD

Next steps:

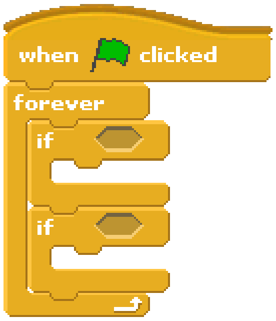
Make the TFT display different text. Make it cycle between displaying different messages.

Project 3: Reading Sensors

Project idea: The on-board sensors on the Sensorian can be accessed in Scratch through Scratch sensor objects.

To get started:

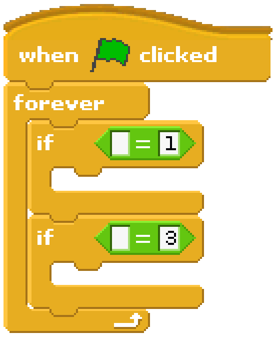
1) Create the car sprite. Remove the default cat sprite by right clicking on it and selecting “delete”. Click on “Choose new sprite from file”, choose “Transportation”, choose “car-blue”, click “OK”.
2) Create the main program template. Click on “Control” and add the “when green flag clicked” to the scripts window. Next, drag the “forever” block onto the scripts window and attach it to the “when green flag clicked” block in the scripts window. Add two “if” blocks to the inside of the “forever” block.



```
when clicked
forever
  if
  if
```

Figure 3: The control structure template for the sensor reading program

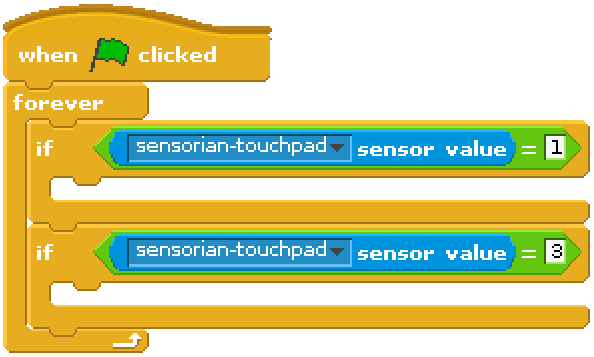
3) Click on “Operators” and add an “=” block to each “if” block. The right side of the first “=” block should be set to **1** while the right side of the second “=” block should be set to **3**.



```
when clicked
forever
  if  = 1
  if  = 3
```

Figure 4: Adding "=" blocks

4) Click on “Sensing” and add a “sensor value” block to the left side of each “=” block and set its sensor to “sensorian-touchpad”.



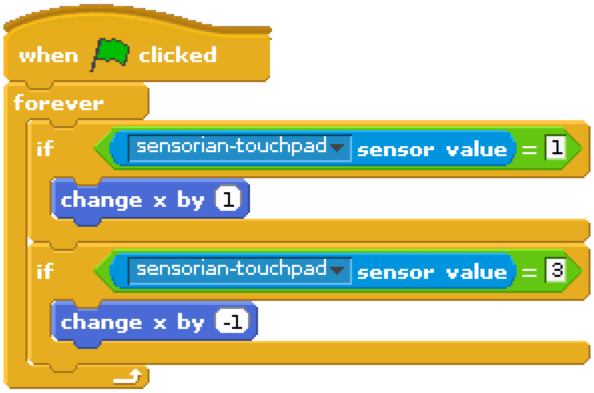
```
when [flag] clicked
forever
    if <(sensorian-touchpad ▾ sensor value) = 1>
    if <(sensorian-touchpad ▾ sensor value) = 3>
```

Figure 5: Adding "sensor value" blocks and setting them to "sensorian-touchpad"

5) Click on “Motion” and add a “change x by” to each “if” block. In the first “change x by” block set its value to **1** and in the second “change x by” block set its value to **-1**.

Final program:

The program should now resemble the following. Pressing Sensorian buttons **1** and **3** should move the car.



```
when [flag] clicked
forever
    if <(sensorian-touchpad ▾ sensor value) = 1>
        change x by 1
    if <(sensorian-touchpad ▾ sensor value) = 3>
        change x by -1
```

Figure 6: The final program. The car moves along the horizontal axis based on the capacitive touch button that is pressed

Next steps:

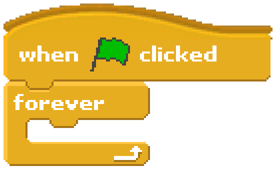
Change the speed of the car. Make the car rotate when it changes direction.

Project 4: Darkness-activated LED

Project idea: The on-board sensors on the Sensorian can be used in a Scratch program to control the outputs on the Sensorian.

To get started:

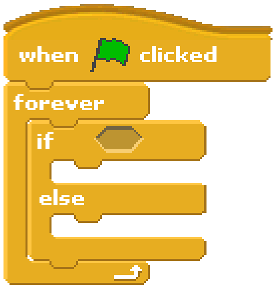
1) Create the "sensorian-led" variable. Click on "Variables", "Make a variable", and name the variable "sensorian-led".
2) Create the main program template. Click on "Control" and add the "when green flag clicked" to the scripts window. Next, drag the "forever" block onto the scripts window and attach it to the "when green flag clicked" block in the scripts window.



```
when 🏁 clicked
forever
```

Figure 7: Creating the main program loop

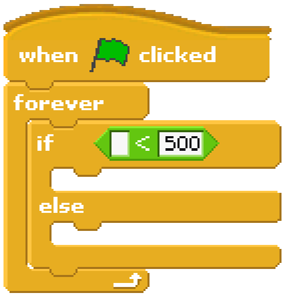
3) Add an "if/else" block to the inside of the "forever" block.



```
when 🏁 clicked
forever
  if
  else
```

Figure 8: Adding the if/else conditional statement block

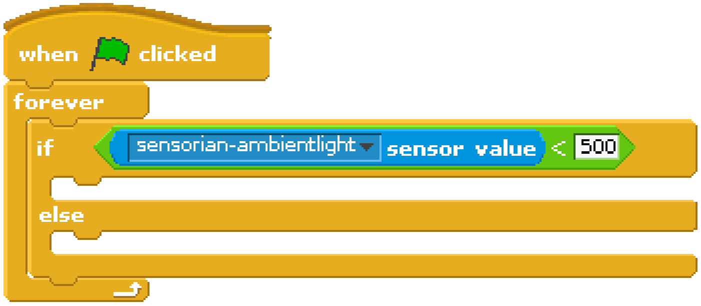
4) Click on "Operators" and add a "<" block to the "if/else" block. Set the right side of the "<" block to 500.



```
when 🏁 clicked
forever
  if  [ ] < 500
  else
```

Figure 9: Adding the "<" operator and comparing to 500

5) Click on “Sensing” and add a “sensor value” block to the left side of the “<” block. Set its sensor to “sensorian-ambientlight”.



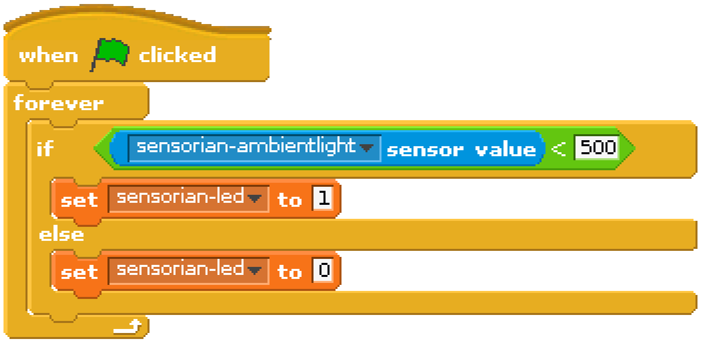
```
when clicked
forever
  if sensorian-ambientlight sensor value < 500
  else
```

Figure 10: Adding the "sensorian-ambientlight" sensor value block

6) Click on “Variables” and add a “set” block to the “if” section of the “if/else” block and another “set” block to the “else” section of the “if/else” block. In the first “set” block, set its variable to “sensorian-led” and its value to **1**. In the second “set” block, set its variable to “sensorian-led” and its value to **0**.

Final program:

The program should now resemble the following. If there is sufficient ambient light, the LED will turn off. If there is not sufficient ambient light, the LED will illuminate.



```
when clicked
forever
  if sensorian-ambientlight sensor value < 500
    set sensorian-led to 1
  else
    set sensorian-led to 0
```

Figure 11: Add the set blocks to control the state of the LED.

Next steps:

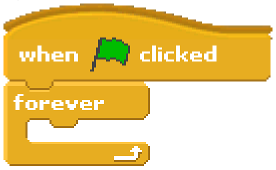
Change the threshold for which the LED illuminates. If the LED is illuminated, change the threshold to compensate for the light emitted from the LED

Project 5: LCD Environment Monitoring Dashboard

Project idea: The TFT LCD display can be used to display sensor data. This is useful for making projects such as a digital weather station. It is also useful for debugging.

To get started:

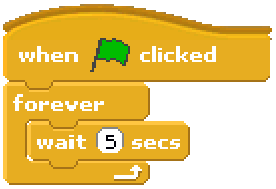
1) Create the "sensorian-tft" variable. Click on "Variables", "Make a variable", and name the variable "sensorian-tft".
2) Create the main program template. Click on "Control" and add the "when green flag clicked" to the scripts window. Next, drag the "forever" block onto the scripts window and attach it to the "when green flag clicked" block in the scripts window.



```
when ⚑ clicked
forever
```

Figure 12: Adding a "forever" block for making the main loop

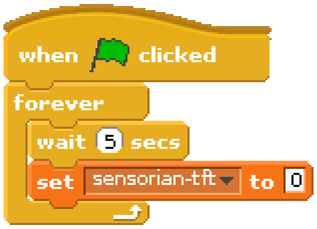
3) Control the refresh rate. Add a "wait" block into the "forever" block. Set its value to **5** seconds.



```
when ⚑ clicked
forever
    wait 5 secs
```

Figure 13: Add a wait block to control the refresh rate of the main loop

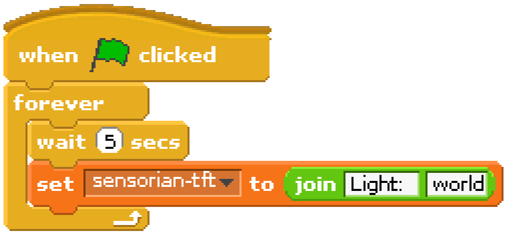
4) Click on "Variables" and insert, below the "wait" block, a "set" block and set its variable to "sensorian-tft".



```
when ⚑ clicked
forever
    wait 5 secs
    set sensorian-tft ▾ to 0
```

Figure 14: Add this set block to clear the TFT-LCD

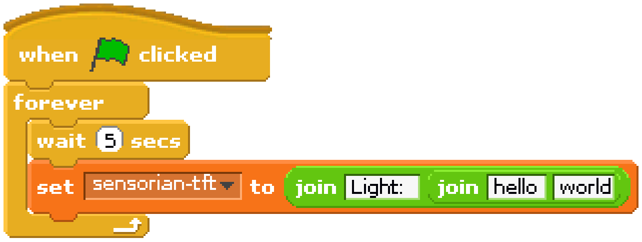
5) Click on "Operators" and insert into the "set" block a "join" block. Into the left side of the "join" block write, "Light: ".



```
when [flag] clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: world
```

Figure 15: Adding the first join block

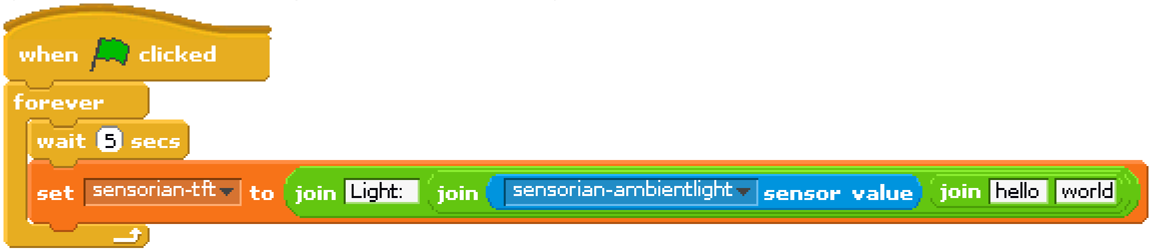
6) Into the right side of the join block insert another join block.



```
when [flag] clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join hello world
```

Figure 16: Adding the second join block

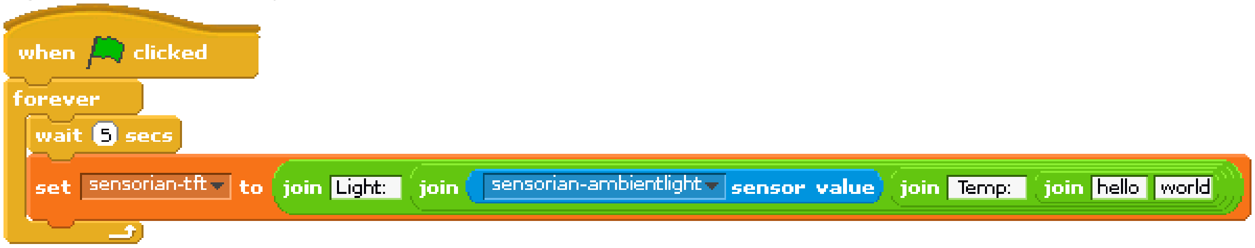
7) Click on “Sensing” and insert a “sensor value” block into the left side of the new “join” block. Set its sensor to “sensorian-ambientlight”. Click on “Operators” and add another “join” block into the right side of the new “join” block.



```
when [flag] clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join hello world
```

Figure 17: Adding a "sensorian-ambientlight" sensor value block into the left side of the new join block, and another join block into the right side

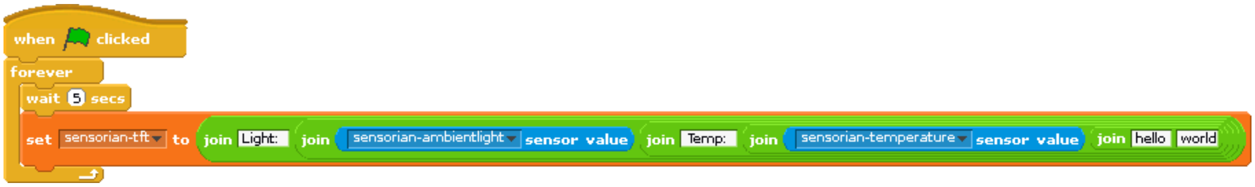
8) Set the left side of the new “join” block to “ Temp: ” and add another join block into the right side of the new join block.



```
when [flag] clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join Temp: join hello world
```

Figure 18: Setting the left side of the new join block to " Temp: " and adding another join block into the right side

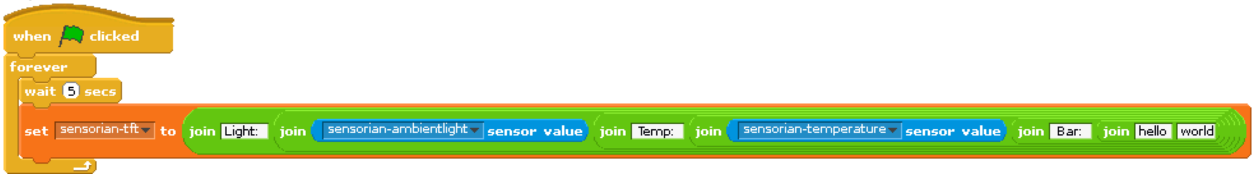
9) Place a “sensor value” block into the left side of the new “join” block and set its sensor to “sensorian-temperature”. Place a new “join” block into the right side of the “join” block.



```
when clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join Temp: join sensorian-temperature sensor value join hello world
```

Figure 19: Adding the new sensor value and join blocks

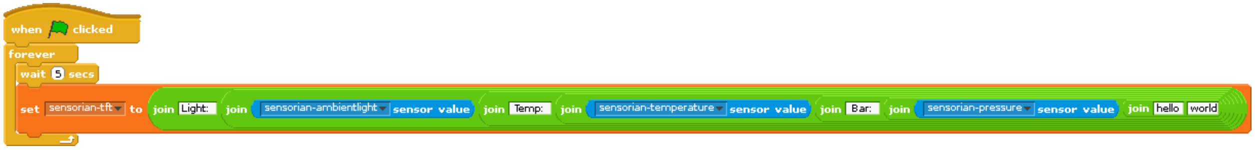
10) Set the left side of the new “join” block to “ Bar: ” and add another “join” block to the right side.



```
when clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join Temp: join sensorian-temperature sensor value join Bar: join hello world
```

Figure 20: Setting the left side of the join block to " Bar: " and adding a new join block

11) Add a “sensor value” block to the left side of the new “join” block and set its sensor to “sensorian-pressure”. Add another “join” block into the right side of the “join” block.



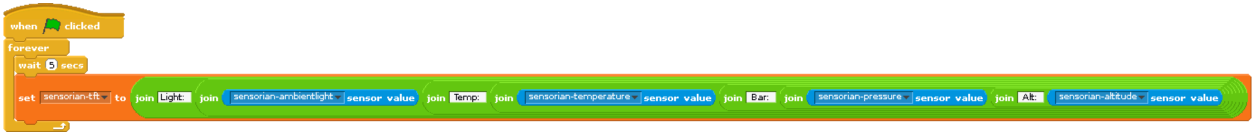
```
when clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join Temp: join sensorian-temperature sensor value join Bar: join sensorian-pressure sensor value join hello world
```

Figure 21: Adding the new sensor value and join blocks

12) Set the left side of the new “join” block to “ Alt: ” and place a “sensor value” block into the right side of the new “join” block. Set the sensor to “sensorian-altitude”.

Final program:

The final program should resemble the following. Adjust the spacing around the title words to make each number print on its own line.



```
when clicked
forever
  wait 5 secs
  set sensorian-tft to join Light: join sensorian-ambientlight sensor value join Temp: join sensorian-temperature sensor value join Bar: join sensorian-pressure sensor value join Alt: sensorian-altitude sensor value
```

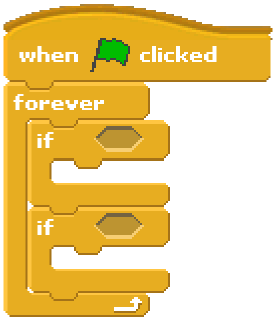
Figure 22: Final program

Project 6: Tilt Sensor

Project idea: The Sensorian can detect the direction in which it is pointing. Scratch can read this information and use it as input for a program.

To get started:

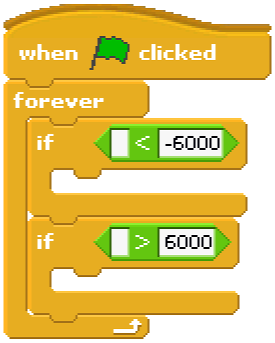
1) Create the LEGO block sprite. Remove the default cat sprite by right clicking on it and selecting "delete". Click on "Choose new sprite from file", choose "Things", choose "lego", and click "OK".
2) Create the main program template. Click on "Control" and add the "when green flag clicked" to the scripts window. Next, drag the "forever" block onto the scripts window and attach it to the "when green flag clicked" block in the scripts window. Add two "if" blocks to the inside of the "forever" block.



```
when clicked
forever
    if
    if
```

Figure 23: Creating the main program control structure template

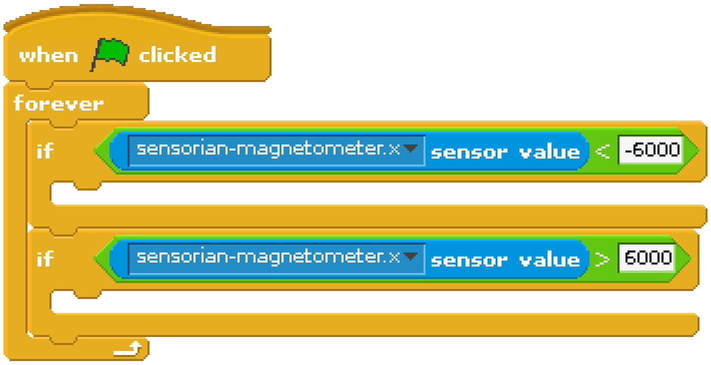
3) Click on "Operators" and add a "<" block to the first "if" block. Set the right side of it to **-6000**. Add a ">" block to the second "if" block. Set the right side of it to **6000**.



```
when clicked
forever
    if  < -6000
    if  > 6000
```

Figure 24: Add the "<" and ">" operators

4) Click on "Sensing" and add a "sensor value" block to the left side of the "<" block and another to the left side of the ">" block. In both cases, set the sensor to "sensorian-magnetometer.x"



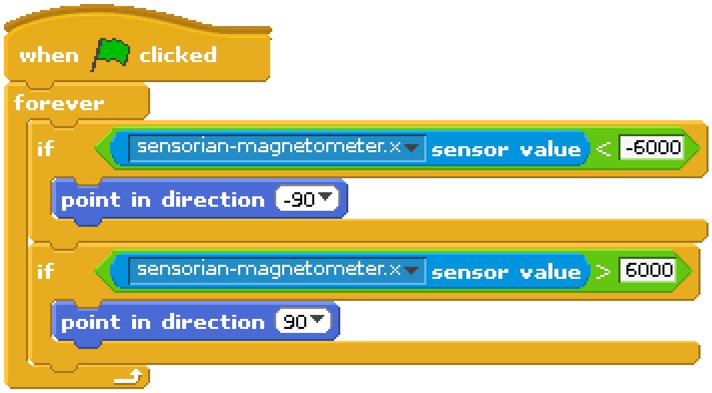
```
when [flag] clicked
forever
  if <(sensorian-magnetometer.x ▾ sensor value) < -6000>
  if <(sensorian-magnetometer.x ▾ sensor value) > 6000>
```

Figure 25: Add the sensor value blocks

5) Click on "Motion" and add a "point in direction" block to both "if" blocks. For the "point in direction" block for the first "if" block, set the direction to "-90". For the "point in direction" block for the second "if" block, set the direction to "90".

Final program:

The final program should resemble the following. Tilt the Raspberry Pi + Sensorian back and forth to see the LEGO block filp accordingly.



```
when [flag] clicked
forever
  if <(sensorian-magnetometer.x ▾ sensor value) < -6000>
    point in direction (-90 ▾)
  if <(sensorian-magnetometer.x ▾ sensor value) > 6000>
    point in direction (90 ▾)
```

Figure 26: Final program

Next steps:

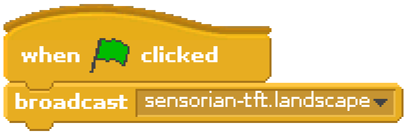
Make a program which uses information from all 3 dimensions of the magnetometer data.

Project 7: Digital TFT-LCD Clock

Project idea: The TFT-LCD works in both portrait and landscape modes. Broadcasting **sensorian-tft.portrait** and **sensorian-tft.landscape** changes the LCD mode to portrait or landscape respectively. The Sensorian also has a Real-Time clock which is accessible through Scratch.

To get started:

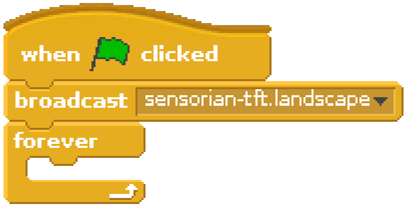
1) Start the LCD display in landscape mode. Click on "Control" and drag a "when green flag clicked" block into the scripts window. Next drag a "broadcast" block and place it immediately below the "when green flag clicked" block. Set its message to "new…", "sensorian-tft.landscape".



```
when 🏁 clicked
broadcast sensorian-tft.landscape ▾
```

Figure 27: Broadcast "sensorian-tft.landscape" to make the TFT-LCD display text in landscape (wide) mode

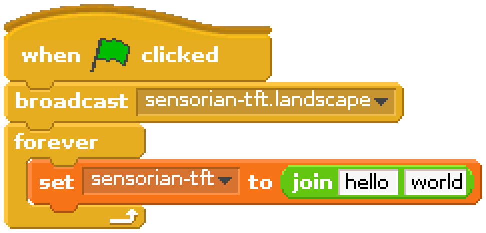
2) Create the main loop. Drag a "forever" block into the scripts window and place it immediately below the "broadcast" block on the scripts window.



```
when 🏁 clicked
broadcast sensorian-tft.landscape ▾
forever
```

Figure 28: Create the main loop

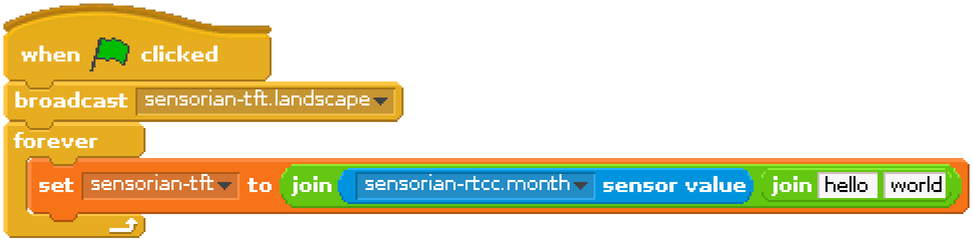
3) Create the "sensorian-tft" variable. Click on "Variables", "Make a variable", and name the variable "sensorian-tft".
4) Place a "set" block in the scripts window inside the "forever" block. Set its variable to "sensorian-tft". Click on "Operators" and add a "join" block into the "set" block as its value.



```
when 🏁 clicked
broadcast sensorian-tft.landscape ▾
forever
    set sensorian-tft ▾ to join hello world
```

Figure 29: Adding the set and join blocks

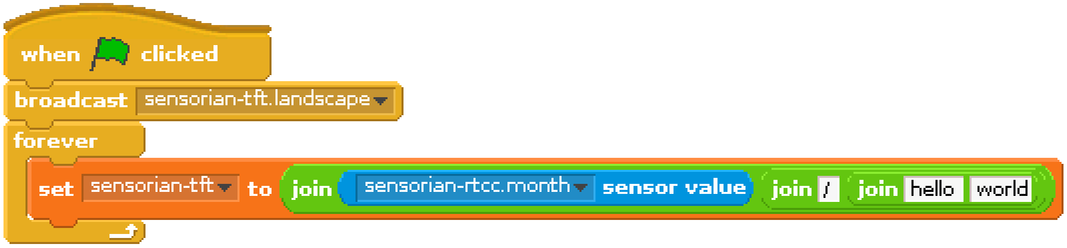
5) Click on “Sensing” and place a “sensor value” block into the left side of the “join” block. Set its sensor to “sensorian-rtcc.month”. Into the right side of the “join” block, place another “join” block.



```
when ⚑ clicked
broadcast sensorian-tft.landscape ▾
forever
    set sensorian-tft ▾ to join ( sensorian-rtcc.month ▾ sensor value ) join hello world
```

Figure 30: Adding the sensor value and join blocks

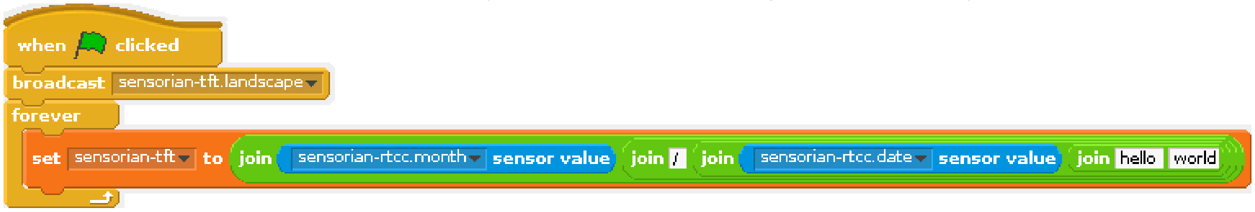
6) Set the left side of the new “join” block to “/” and add another “join” block into the right side of the “join” block.



```
when ⚑ clicked
broadcast sensorian-tft.landscape ▾
forever
    set sensorian-tft ▾ to join ( sensorian-rtcc.month ▾ sensor value ) join / join hello world
```

Figure 31: Setting the left side of the new join block and adding another join block to its right side

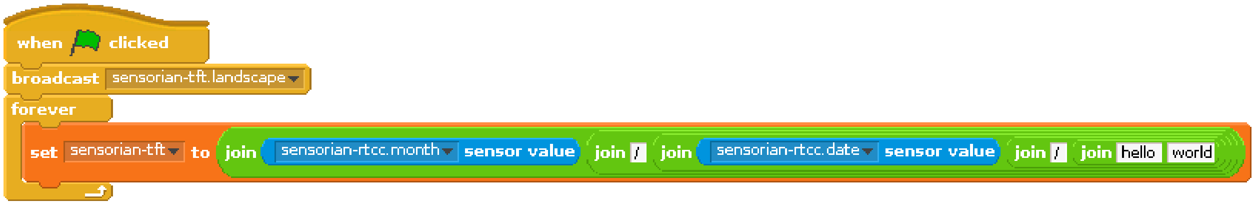
7) Into the left side of the new “join” block, place a “sensor value” block and set its sensor to “sensorian-rtcc.date” Place a new “join” block into the right side of the “join” block.



```
when ⚑ clicked
broadcast sensorian-tft.landscape ▾
forever
    set sensorian-tft ▾ to join ( sensorian-rtcc.month ▾ sensor value ) join / join ( sensorian-rtcc.date ▾ sensor value ) join hello world
```

Figure 32: Adding the sensor value and join blocks

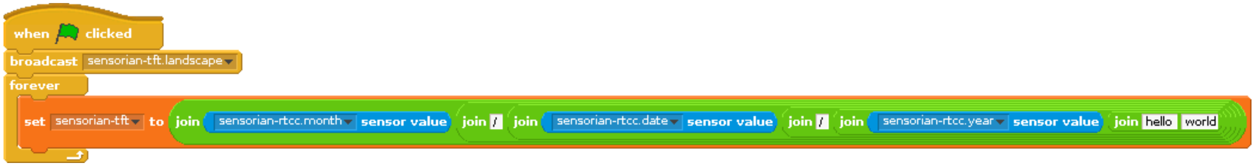
8) Set the left side of the new “join” block to “/” and add another “join” block into the right side of the “join” block.



```
when ⚑ clicked
broadcast sensorian-tft.landscape ▾
forever
    set sensorian-tft ▾ to join ( sensorian-rtcc.month ▾ sensor value ) join / join ( sensorian-rtcc.date ▾ sensor value ) join / join hello world
```

Figure 33: Setting the left side of the new join block and adding another join block to its right side

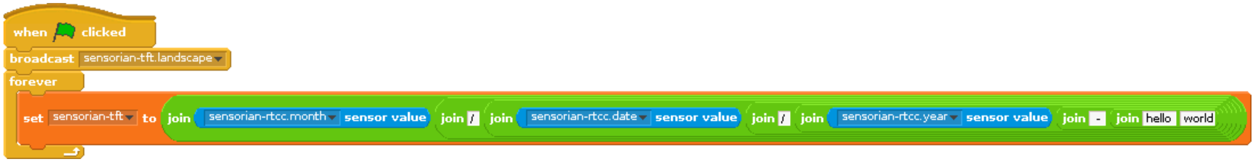
9) Into the left side of the new “join” block add a “sensor value” block and set its sensor to “sensorian-rtcc.year”. Into the right side of the “join” block place another “join” block.



```
when clicked
broadcast sensorian-tft.landscape
forever
  set sensorian-tft to join sensorian-rtcc.month sensor value join / join sensorian-rtcc.date sensor value join / join sensorian-rtcc.year sensor value join hello world
```

Figure 34: Adding the sensor value and join blocks

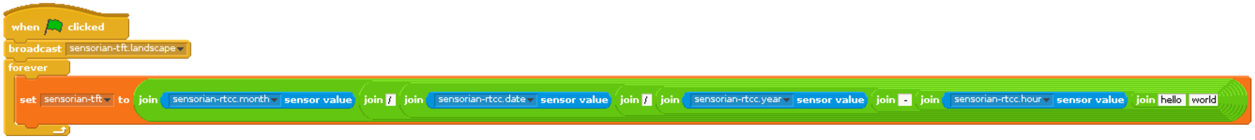
10) Set the left side of the new “join” block to “ – “ and add another “join” block into the right side.



```
when clicked
broadcast sensorian-tft.landscape
forever
  set sensorian-tft to join sensorian-rtcc.month sensor value join / join sensorian-rtcc.date sensor value join / join sensorian-rtcc.year sensor value join - join hello world
```

Figure 35: Setting the left side of the new join block and adding another join block to its right side

11) Add a “sensor value” block to the left side of the new “join” block and set its sensor to “sensorian-rtcc.hour”. Add another “join” block into the right side of the “join” block.



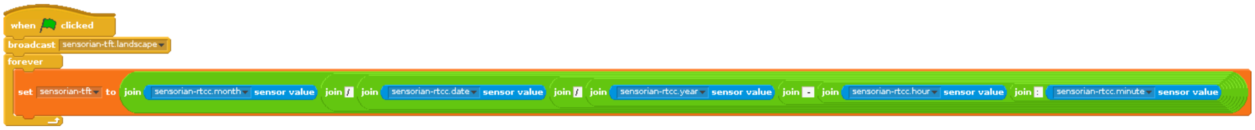
```
when clicked
broadcast sensorian-tft.landscape
forever
  set sensorian-tft to join sensorian-rtcc.month sensor value join / join sensorian-rtcc.date sensor value join / join sensorian-rtcc.year sensor value join - join sensorian-rtcc.hour sensor value join hello world
```

Figure 36: Adding the sensor value and join blocks

12) Set the left side of the new “join” block to “:” and insert a “sensor value” block into the right side. Set its sensor to “sensorian-rtcc.minute”

Final Program:

The final program should resemble the following. Running it should cause the date and time (to the nearest minute) to be printed on the TFT-LCD.



```
when clicked
broadcast sensorian-tft.landscape
forever
  set sensorian-tft to join sensorian-rtcc.month sensor value join / join sensorian-rtcc.date sensor value join / join sensorian-rtcc.year sensor value join - join sensorian-rtcc.hour sensor value join : sensorian-rtcc.minute sensor value
```

Figure 37: Final program

Next steps:

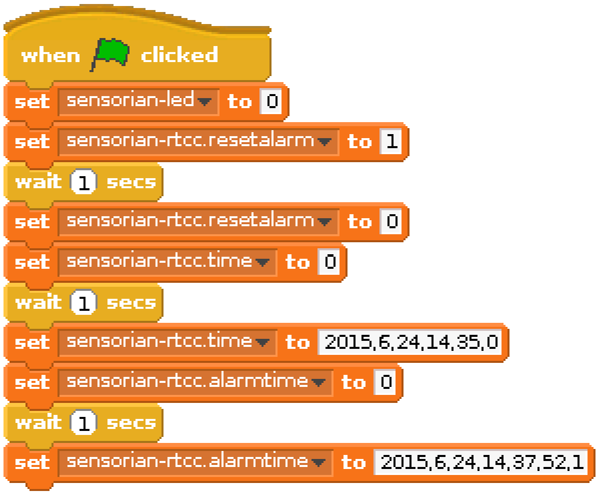
Use the magnetometer to make the text rotate whenever the Sensorian is rotated.

Project 8: Alarm Clock

Project idea: The Sensorian Real-Time clock has alarm functionality. Developers can set the clock time, set the alarm time, and check alarm status all from Scratch.

To get started:

1) Create the needed variables. Click on “Variables”, “Make a variable”, and name the variable “sensorian-led”. Repeat this process for all other variables, “sensorian-rtcc.resetalarm”, “sensorian-rtcc.time”, and “sensorian-rtcc.alarmtime”.
2) Initialize the program. Click on “Control” and drag a “when green flag clicked” block into the scripts window. Click on “Variables” and add a “set” block immediately below the “when green flag clicked” block. Set its variable to “sensorian-led” and its value to “0”. Add another “set” block and set its variable to “sensorian-rtcc.resetalarm” and its value to “1”. Click on “Control” and add a “wait” block below the “set” blocks. Set its time to “1”. Click on “Variables” and add another “set” block to the script. Set its variable to “sensorian-rtcc.resetalarm” and its value to “0”. Add another “set” block and set its variable to “sensorian-rtcc.time” and its value to “0”. Add another “wait” block and set its time to “1”. Add another “set” block and set its variable to “sensorian-rtcc.time” and its value to “2015,6,24,14,35,0”. This will set the Real-Time clock to June 24th 2015 at 14:35:00. Add another “wait” block and set its time to “1”. Below the “wait” block and a “set” block and set its variable to “sensorian-rtcc.alarmtime” and its value to “2015,6,24,14,37,52,1”. This will set the alarm in the Real-Time clock to June 24th 2015 at 14:37:52 in *minutes-match* mode.

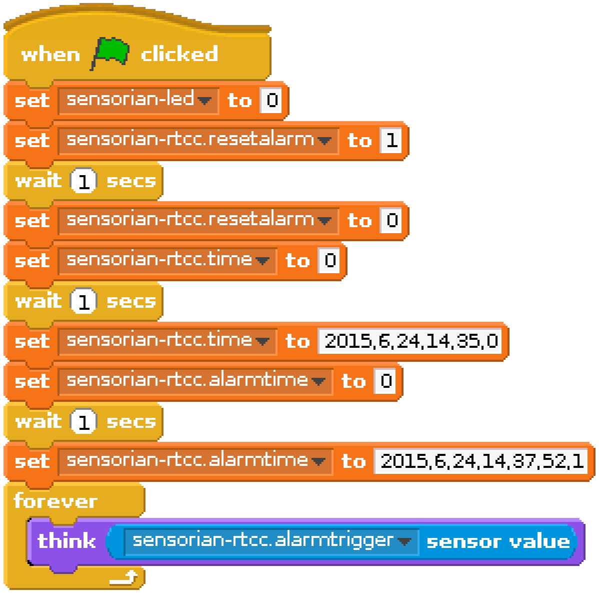


```
when clicked
set sensorian-led to 0
set sensorian-rtcc.resetalarm to 1
wait 1 secs
set sensorian-rtcc.resetalarm to 0
set sensorian-rtcc.time to 0
wait 1 secs
set sensorian-rtcc.time to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime to 0
wait 1 secs
set sensorian-rtcc.alarmtime to 2015,6,24,14,37,52,1
```

Figure 38: Program initialization sequence

3) Click on “Control” and add a “forever” block to the “Scripts” window. Click on “Looks” and a “think” block to the inside of the “forever” block. Click on “Sensing” and add a

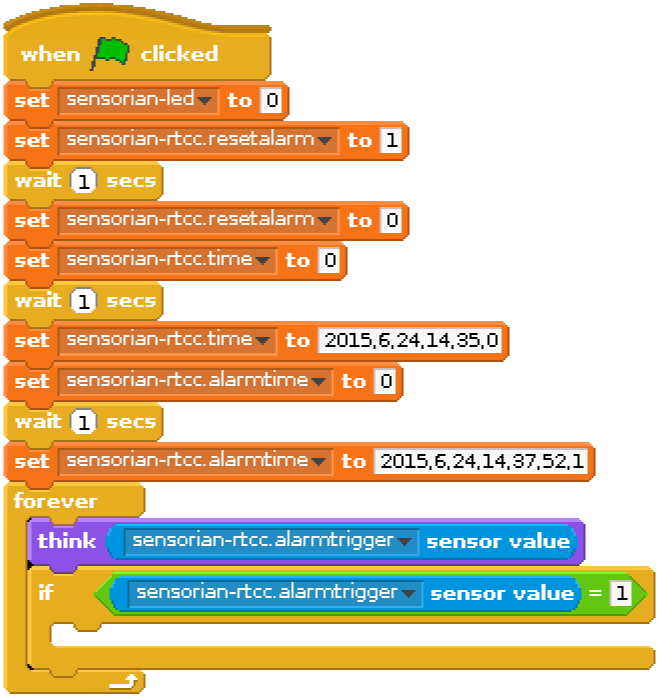
"sensor value" block to the inside of the "think" block. Set its sensor to "sensorian-rtcc.alarmtrigger". This is useful for debugging.



```
when [flag] clicked
set sensorian-led to 0
set sensorian-rtcc.resetalarm to 1
wait 1 secs
set sensorian-rtcc.resetalarm to 0
set sensorian-rtcc.time to 0
wait 1 secs
set sensorian-rtcc.time to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime to 0
wait 1 secs
set sensorian-rtcc.alarmtime to 2015,6,24,14,37,52,1
forever
    think (sensorian-rtcc.alarmtrigger sensor value)
```

Figure 39: Create the main loop and add a think block for debugging

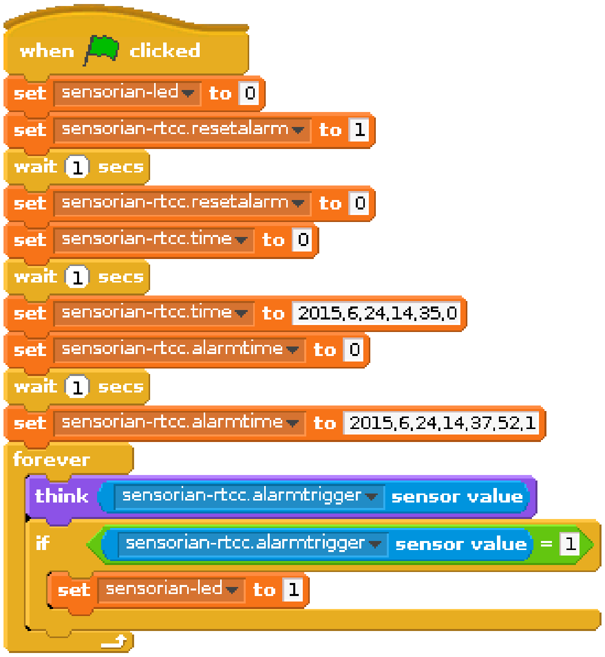
4) Click on "Control" and add an "if" block below the "think" block. Click on "Operators" and add a "=" block to the conditional part of the "if" block. Set the right side to "1". Click on "Sensing" and add a "sensor value" block to the left side of the "=" block. Set its sensor to "sensorian-rtcc.alarmtrigger".



```
when [flag] clicked
set sensorian-led to 0
set sensorian-rtcc.resetalarm to 1
wait 1 secs
set sensorian-rtcc.resetalarm to 0
set sensorian-rtcc.time to 0
wait 1 secs
set sensorian-rtcc.time to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime to 0
wait 1 secs
set sensorian-rtcc.alarmtime to 2015,6,24,14,37,52,1
forever
    think (sensorian-rtcc.alarmtrigger sensor value)
    if <(sensorian-rtcc.alarmtrigger sensor value) = 1>
```

Figure 40: Using an if block to run code if the alarm is triggered

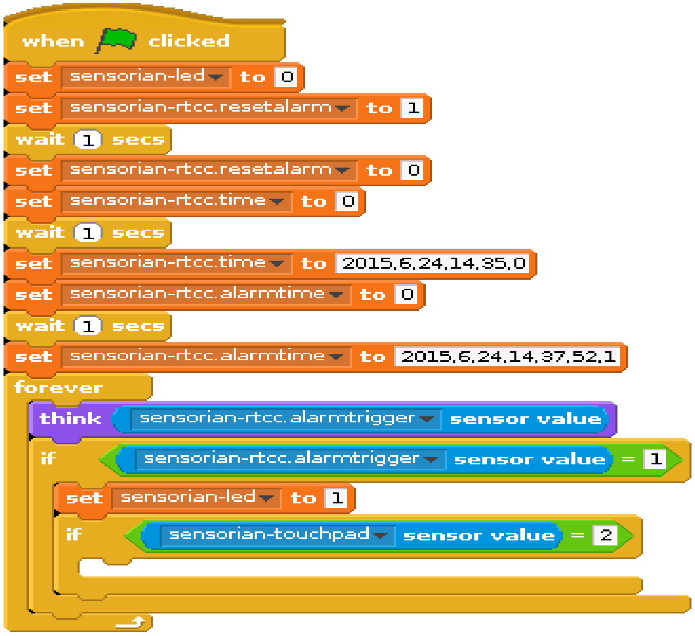
5) Click on “Variables” and add a “set” block to the inside of the “if” block. Set the variable to “sensorian-led” and the value to “1”.



```
when ⚑ clicked
set sensorian-led ▾ to 0
set sensorian-rtcc.resetalarm ▾ to 1
wait 1 secs
set sensorian-rtcc.resetalarm ▾ to 0
set sensorian-rtcc.time ▾ to 0
wait 1 secs
set sensorian-rtcc.time ▾ to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime ▾ to 0
wait 1 secs
set sensorian-rtcc.alarmtime ▾ to 2015,6,24,14,37,52,1
forever
    think (sensorian-rtcc.alarmtrigger ▾ sensor value)
    if <(sensorian-rtcc.alarmtrigger ▾ sensor value) = 1>
        set sensorian-led ▾ to 1
```

Figure 41: Add the set block

6) Click on “Control” and add an “if” block to the inside of the other “if” block. Click on “Operators” and add an “=” block to the condition part of the “if” block. Set its right side to “2”. Click on “Sensing” and add a “sensor value” block to the left side of the “=” block. Set its sensor to “sensorian-touchpad”.



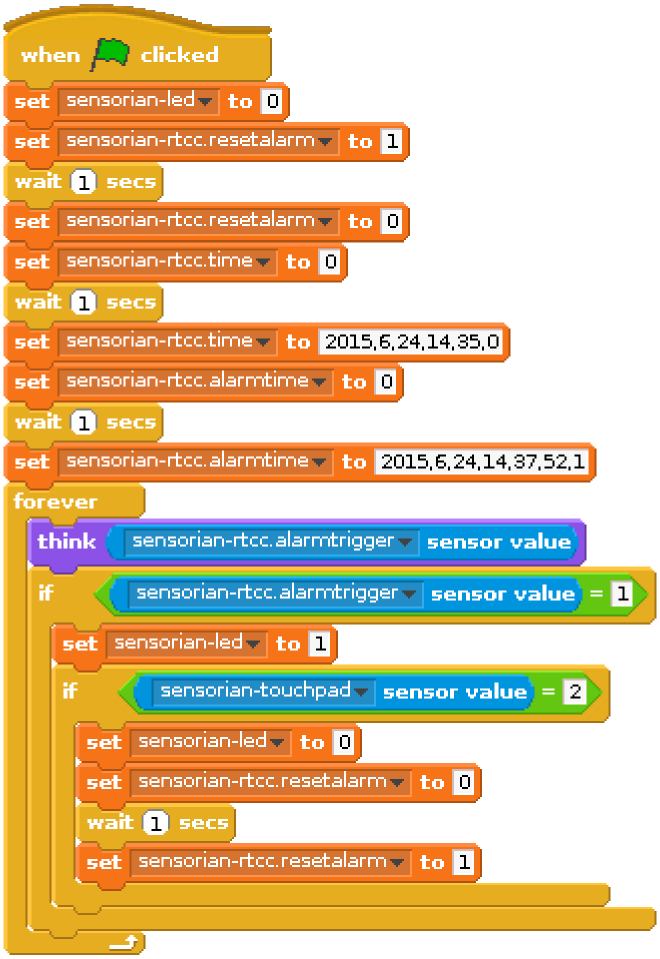
```
when ⚑ clicked
set sensorian-led ▾ to 0
set sensorian-rtcc.resetalarm ▾ to 1
wait 1 secs
set sensorian-rtcc.resetalarm ▾ to 0
set sensorian-rtcc.time ▾ to 0
wait 1 secs
set sensorian-rtcc.time ▾ to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime ▾ to 0
wait 1 secs
set sensorian-rtcc.alarmtime ▾ to 2015,6,24,14,37,52,1
forever
    think (sensorian-rtcc.alarmtrigger ▾ sensor value)
    if <(sensorian-rtcc.alarmtrigger ▾ sensor value) = 1>
        set sensorian-led ▾ to 1
        if <(sensorian-touchpad ▾ sensor value) = 2>
```

Figure 42: Add an if block to run code if B2 is pressed while the alarm is triggered

7) Click on “Variables” and add a “set” block to the inside of inner “if” block. Set its variable to “sensorian-led” and its value to “0”. Add another “set” block and set its variable to “sensorian-rtcc.resetalarm” and its value to “0”. Click on “Control” and add a “wait” block and set its time to “1”. Click on “Variables” and add a “set” block. Set its variable to “sensorian-rtcc.resetalarm” and its value to “1”.

Final program:

The final program should resemble the following. Clicking on the green flag will cause the LED to illuminate after two minutes. At that point, pressing capacitive touch button “B2” will turn the LED off.



```
when 🟩 clicked
set sensorian-led to 0
set sensorian-rtcc.resetalarm to 1
wait 1 secs
set sensorian-rtcc.resetalarm to 0
set sensorian-rtcc.time to 0
wait 1 secs
set sensorian-rtcc.time to 2015,6,24,14,35,0
set sensorian-rtcc.alarmtime to 0
wait 1 secs
set sensorian-rtcc.alarmtime to 2015,6,24,14,37,52,1
forever
    think sensorian-rtcc.alarmtrigger sensor value
    if sensorian-rtcc.alarmtrigger sensor value = 1
        set sensorian-led to 1
        if sensorian-touchpad sensor value = 2
            set sensorian-led to 0
            set sensorian-rtcc.resetalarm to 0
            wait 1 secs
            set sensorian-rtcc.resetalarm to 1
```

Figure 43: Final program

Next steps:

Make the program play a sound when the alarm is triggered.

Make the LED flash when the alarm is triggered.

Make the TFT-LCD display the amount of time remaining before the alarm is triggered.