Engduino Application Documentation

My Engduino application for the C coursework consists of two programs: one written in Arduino-Engduino IDE and one written in Processing IDE. It takes sensor values from Engduino and displays them in several ways, in real-time.

The application written in Arduino-Engduino uses C/C++ as a programming language and, once uploaded to the Engduino, it takes several data from it like temperature, light level, accelerometer data and magnetometer data, using special libraries. My program takes only one type of data at one time. The code switches between the types of values which are to be displayed by pressing the Engduino button. All of these are sent through the serial port in the laptop/PC, being preceded by a code which tells what sensor was used. Moreover, there is a light show using the LEDs which moves at the rate at which the values are printed.

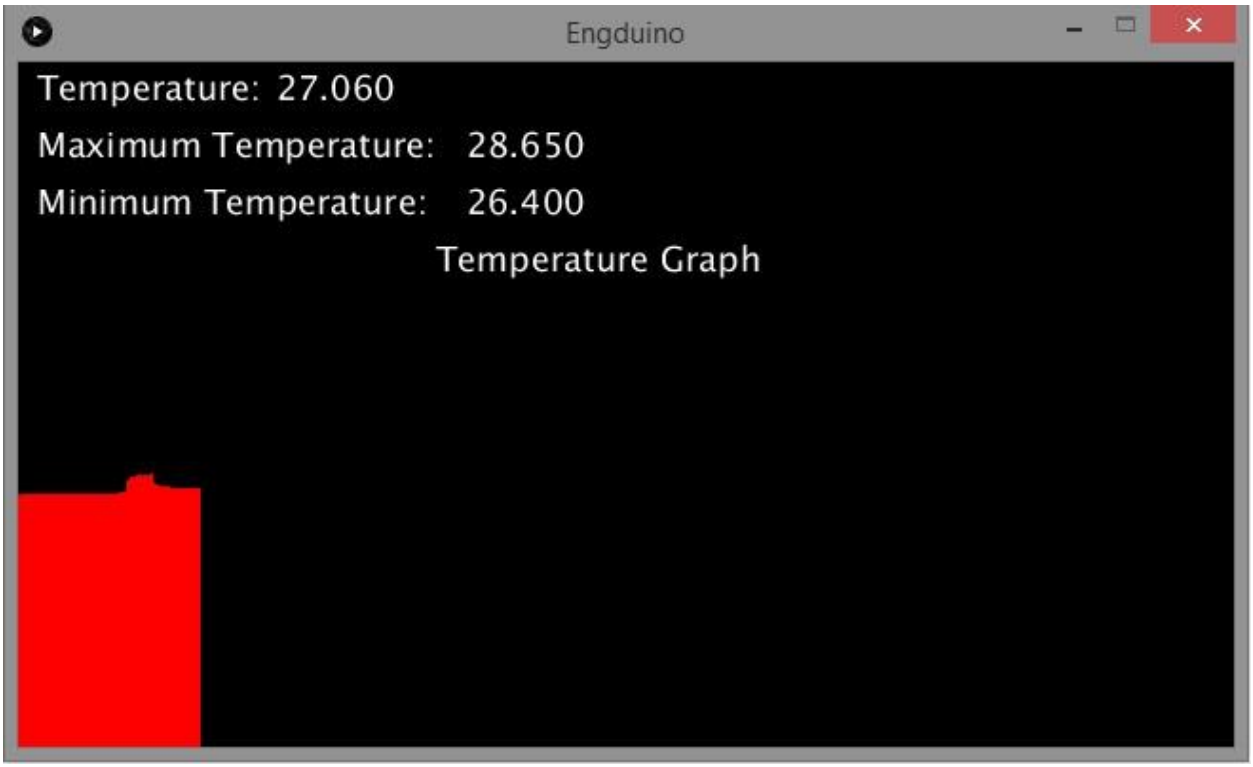
The application in Processing IDE is more complex. Firstly, it looks at which port the Engduino is printing the data and then starts to read this data. After that, by the preceding code received from Engduino, it knows what sensor was used. Moreover, it reads the desired values and displays graphs and values to the screen, corresponding to a certain sensor.

My Processing program has 4 main displays:

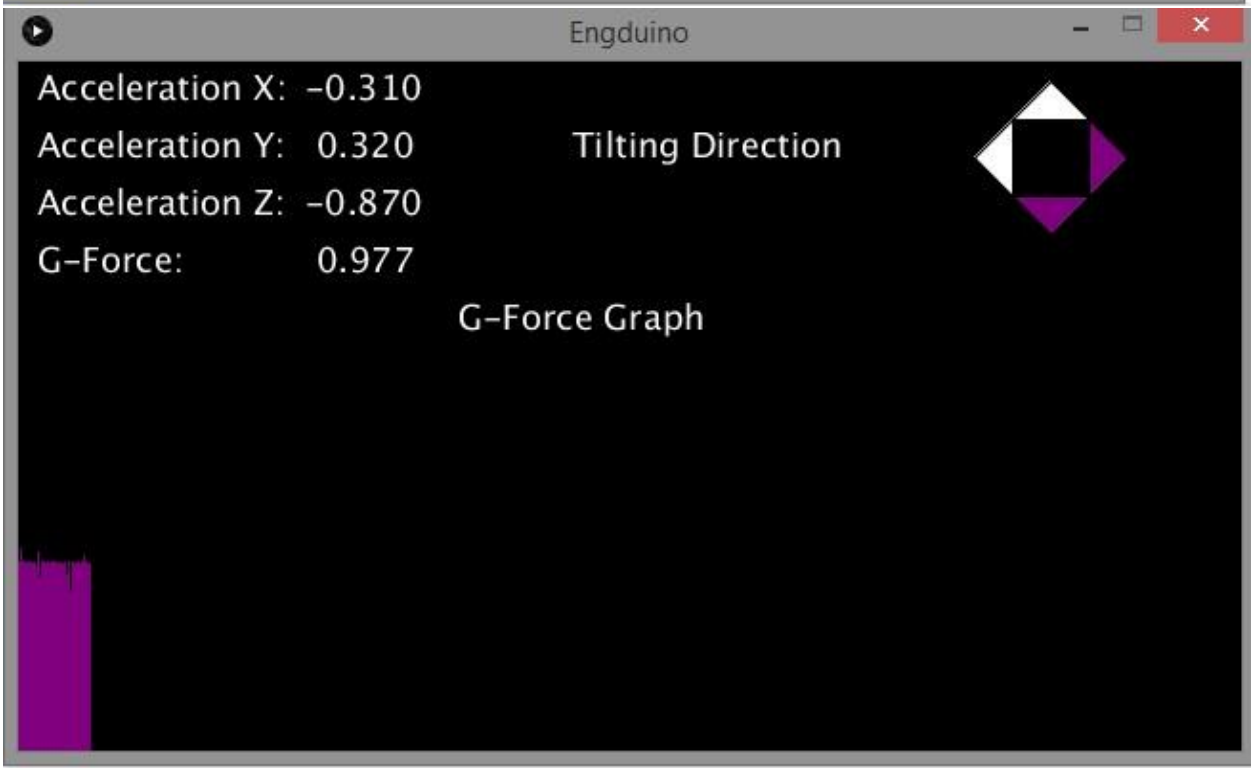
- **Temperature (executed when data from the temperature sensor is received) which displays a real-time graph of the temperature, the actual temperature and the minimum/maximum temperature.**
- **Light (executed when data from the light sensor is received) which displays a real-time graph of the light level, the actual light level and the minimum/maximum light level. You can see that the LEDs from Engduino slightly influence the sensor.**
- **Accelerometer (executed when data from the accelerometer sensor is received) which displays a real-time graph of the G-Force and the x, y, z values of the acceleration. Moreover, it has arrows which display in which direction the Engduino is tilted.**
- **Magnetometer (executed when data from the magnetometer sensor is received) which displays a real-time graph of the magnitude and the x, y, z values of the magnetic field. Moreover, it has rectangle which lights when the Engduino is pointed to the geographic North. NOTE!!!: The Engduino sensor has to be calibrated in order for it to display North. In order to do this, slowly rotate the Engduino when you have the Processing program open in the magnetometer mode (or the laptop/PC if you do not have an extension cable) 360 degrees. After that, it will show North.**

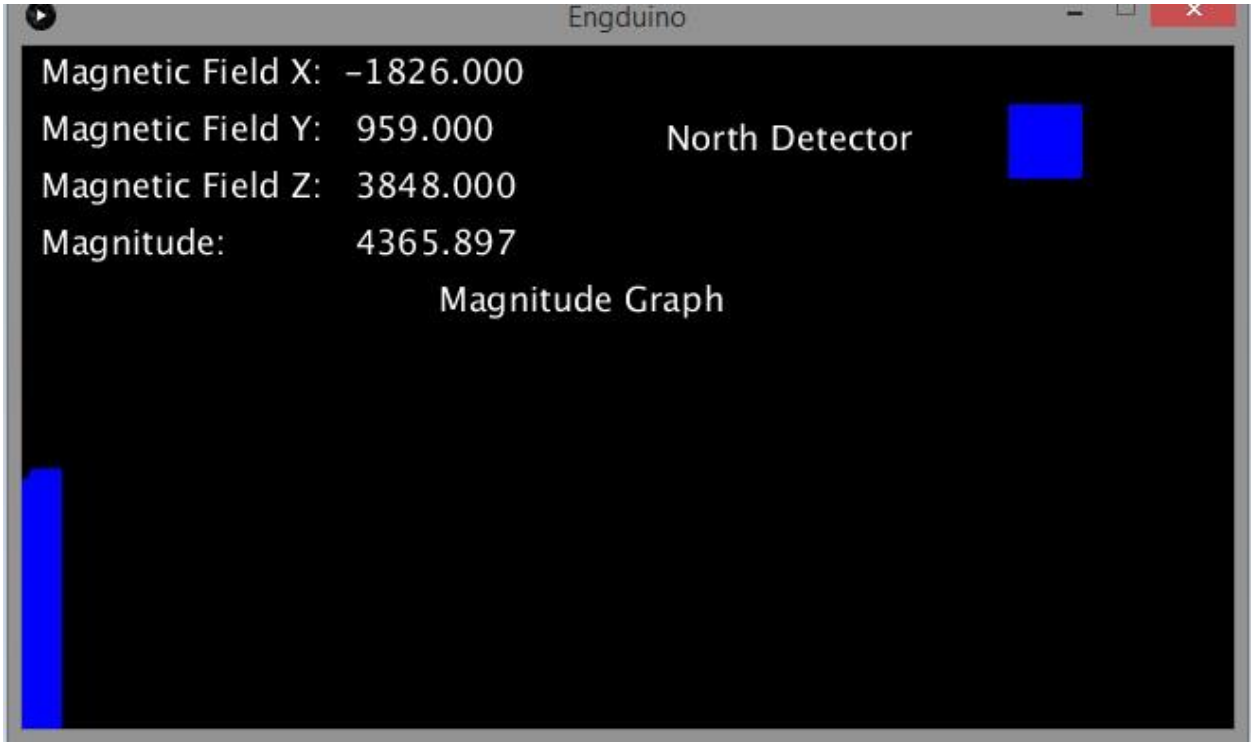
HOW TO SET-UP

1) **Open Arduino-Engduino IDE(if you do not have one installed, download it from https://engduino.org/)**
2) **In the Arduino-Engduino IDE, open 'Engduino.ino', located in 'Engduino/Arduino-Engduino_IDE/Engduino/'**
3) **Upload the code to an Engduino**
4) **Open Processing IDE (if you do not have one installed, download it from https://processing.org/)**
5) **In the Processing IDE, open 'Engduino.pde', located in 'Engduino/Processing_IDE/Engduino/'**
6) **Connect the Engduino to a serial port(it would be better if the other ports are disconnected)**
7) **Start Engduino**
8) **Run the code in Processing IDE**
9) **Play with the application using the button and tilting like suggested in the description. Do not forget to calibrate the magnetometer. In order to do this, slowly rotate the Engduino when you have the Processing program open in the magnetometer mode (or the laptop/PC if you do not have an extension cable) 360 degrees. After that, it will show North.**




Engduino
Light Level: 72
Maximum Light Level: 638
Minimum Light Level: 6
Light Graph


Engduino
Acceleration X: -0.310
Acceleration Y: 0.320
Acceleration Z: -0.870
G-Force: 0.977
Tilting Direction
G-Force Graph


Engduino
Magnetic Field X: -1826.000
Magnetic Field Y: 959.000
Magnetic Field Z: 3848.000
Magnitude: 4365.897
North Detector
Magnitude Graph