OpenModelica

Open Modelica is a free and [open source](https://en.wikipedia.org/wiki/Open_source) environment based on the [Modelica](https://en.wikipedia.org/wiki/Modelica) modeling language for modeling, simulating, optimizing and analyzing complex dynamic systems. OpenModelica is used in academic and industrial environments. Industrial applications include the use of OpenModelica along with proprietary software in the fields of power plant optimization, automotive and water treatment.

Scope

Open Modelica is a powerful tool that can be used to design and simulate complete control systems. Our project tries to interface it with Arduino by calling c functions from openmodelica. Modelica functions are written in open modelica and they call c functions which give insttructions to Arduino. These codes can be run to perform operations on dc motor, servo motor, led, ldr, thermistor and potentiometer.

1. Downloading OpenModelica

OpenModelica can be downloaded free of cost from <https://openmodelica.org>.

* For Windows

It can be downloaded as an .exe file and the installation wizard should be run to complete the installation. It can be launched by double clicking on the shortcut.

* For Linux

It can be downloaded from <https://openmodelica.org/download/download-linux> and the instructions given on this site must clearly be followed. It can be launched by typing OMEdit on the terminal.

Interfacing with windows.

1. Compiling Serialcomm.c

This can be found in \*some path\* .

To compile this and create a shared object file open command prompt and go to the path where this file is saved.

Write the following commands on cmd

Gcc –c Serialcomm.c

Gcc –shared -o SerialComm.dll Serialcomm.o

2)

Loading the SerialCommunication library.

Open Open Modelica and go to File->Open Model .

Browse to the \*path of user code\* and load it.

\*To set the port in Windows go to device manager->Ports->Advanced and select any port(in this project we have selected port 2). For linux check the port by typing \* /ls…\* on the terminal. If it says ACM0 then the port in 0 and we must take care to pass 0 in the open\_serial function.

Interfacing with LED

1. In the first experiment, we will light up the blue LED on the shield.

For this we need to import the Arduino.SerialCommunication package and call the functions that were written in that package via an object.

2

sComm.open\_serial(1,2,115200)\* opens the com port 2 for serial communication.

If 0 is returned and the open operation was unsuccessful and an appropriate messsge is printed.

Else sComm.cmd\_digital\_out(1,9,1) sends a high on pin 9 which corresponds to the lighting up of the blue LED.

3 sComm.open\_serial(1) closes the serial port after lighting up the LED.

1