Lab session week 4 and 5: IoT Programming with Arduino – Analog signals and edgeserver connection

Aim

The aim of this tutorial is for students to be able to confidently program the Arduino board when using analog signals and also guides students to set up their edge server, and connectit to the Arduino board via serial communication.

Important Information

- Please do not power the sensor nodes (Arduino) before getting the confirmation from your tutor.
- Please use your own laptop.
- If your laptop does not have a USB type A socket, you will need to bring an adaptor.
- Be gentle with the hardware.
- Do not connect your laptop or a physical Raspberry Pi via Ethernet to the Swinburne network.

Task 1: Controlling the speed of a DC motor using a potentiometer

This task is aimed at helping you to work with analog signals. In this first task you will be controlling the speed of a DC motor using a potentiometer.

1. Make connections as shown in Figure 1.
2. Make sure your circuit is correct. Pay attention to the wires, PIN connections, resistor, polarity of the LED, etc. Ask your tutor for help if required. **Please DO NOT power ON the Arduino board yet.**
3. Next, open Arduino IDE (installed in your computer).
4. Download the “Week4&5 Resources” folder from canvas Week4 Modules. Open the folder named “Task 1”. From your Arduino IDE, open the file “Task1.ino” saved in your recently opened “Task 1” Folder.
5. Now, connect the Arduino board using the USB cable to your computer which is running Arduino IDE.
6. Before you compile and run, follow the below steps to select the board and port:
    - Select board: Go to Tools menu on your Arduino IDE -> Board-> Select “Arduino Uno”
    - Select port: Go to Tools menu on your Arduino IDE -> Port-> Select the port to which your Arduino Uno is connected.

Please note: The port number might vary for each computer. So select the one that is applicable to you.

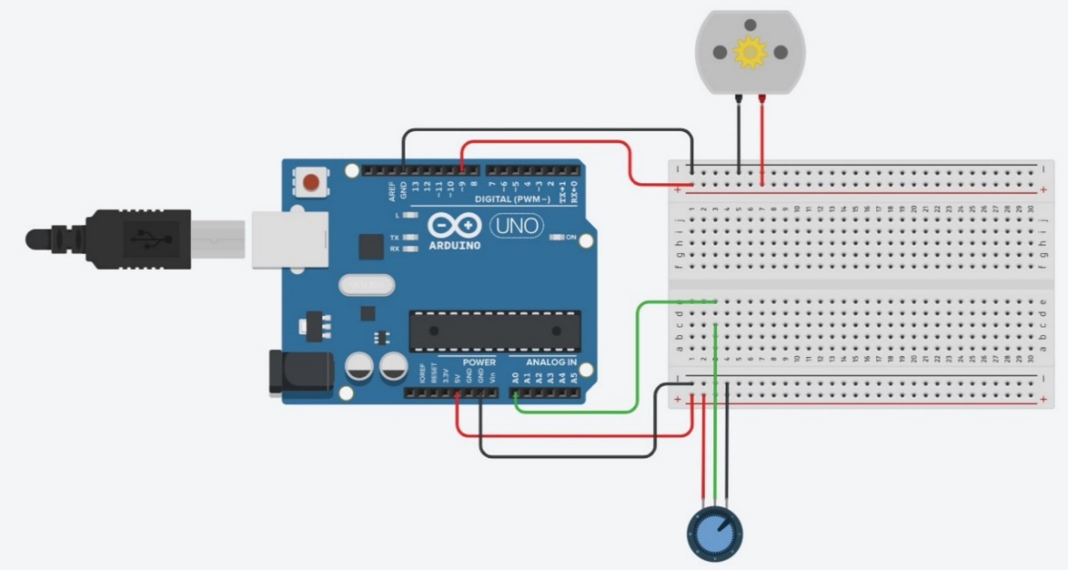

Figure 1: Task 1 circuit

7. Next, verify (compile) the code using sketch drop-down menu (or verify button tick symbol). Upload the code to your Arduino board using the “Upload” option from the “Sketch” drop-down menu or using the right arrow icon on the Arduino IDE.
8. You should now see the LED blinking if your circuit is wired correctly and your code is correct.

Please ask your tutor for help if this doesn’t work.

You can also refer to the demo of this in the “Task 1 Demo.gif” file saved in the Resources folder.

For you to do!!

Task 1.1: Controlling the brightness of an LED using a potentiometer:

Modify your circuit and program to use the potentiometer to control the brightness of an LED instead of a DC motor. Make sure the potentiometer is connected to an analog pin, and the outputs to pulse width modulation (PWM) pins.

Task 2: Serial Communication

In this task you will learn to write a simple program on Arduino IDE to print via serial port "IOT Programming is GREAT!" at a baud rate of 9600. *Refer to the video file "Task 2 Demo.mp4" in the resources folder for video instructions on doing this task.*

1. Open Arduino IDE (installed in your computer).
2. Download the "Week 4&5 Resources" folder from canvas Week4 Modules. Open the folder named "Task 2". From your Arduino IDE, open the file "Task2.ino" saved in your recently opened "Task 2" Folder.
3. Now, connect the Arduino board using the USB cable to your computer which is running Arduino IDE.
4. Before you compile and run, follow the below steps to select the board and port:
    - Select board: Go to Tools menu on your Arduino IDE -> Board-> Select "Arduino Uno"
    - Select port: Go to Tools menu on your Arduino IDE -> Port-> Select the port to which your Arduino Uno is connected.

Please note: The port number might vary for each computer. So select the one that is applicable to you.

5. Next, verify (compile) the code using sketch drop-down menu (or verify button tick symbol). Upload the code to your Arduino board using the "Upload" option from the "Sketch" drop-down menu or using the right arrow icon on the Arduino IDE.
6. Open Serial Monitor from the tools tab on the Arduino IDE i.e., Tools->Serial Monitor. You should now see the message – "IOT Programming is GREAT!" printed on the monitor if your program is correct.
7. Go through the code to understand how to send data to the serial port.

Task 3: Edge Server

Refer to the video file "Task 3 Demo.mp4" in the resources folder for video instructions to do this task.

Now that you are comfortable with the Arduino board, it's time to connect it to the edge server (Raspberry Pi). The first step is to get the server ready by installing the operating system: Raspberry Pi OS (Raspbian). Raspbian is a Debian-based (GNU/Linux) operating system for the Raspberry Pi. You have two options to install the edge server (choose only one).

a) **Virtual Raspberry Pi (Preferred):**

If you want to use a virtual Raspberry Pi, you can install Raspberry Pi OS as avirtual machine in your personal computer/laptop. *Refer to the video file "Task 3 Demo.mp4" in the resources folder for video instructions to do Install a virtual Raspberry Pi on your personal computer/laptop and configure it.*

Alternatively, we have VMWare installed on the computers in your lab tutorials. *Refer to the file "Instructions for VMWare access on Lab Desktops" file in the resources folder for instructions to install/access a virtual Raspberry Pi on the Lab Desktop and configure it.*

b) **Physical Raspberry Pi (Optional):**

If you have a Raspberry Pi, you can install Raspberry Pi OS using Raspberry Pi Imager following the official documentation: https://www.raspberrypi.org/software/

1. Install VirtualBox (or your favourite virtualisation software) in your computer: https:
2. //www.virtualbox.org/wiki/Downloads
3. Download Raspberry Pi Desktop: https://www.raspberrypi.org/software/raspberry-pi-desktop/
4. Install Raspbian on the Virtual Machine. If needed, this guide provides all the installation steps: https://roboticsbackend.com/install-raspbian-desktop-on-a-virtual-machine-virtualbox/

After the installation of the virtual machine, your machine will have NAT Network Adapter. It will be better if you use Bridged Adapter because it will provide your edge server a local IP address from your router. However, if you are on Eduroam network, you won't have internet access so you will need to use NAT.

Settings -> Network -> Attached to: Bridged Adapter

Settings -> Network -> Attached to: NAT (If you are on Eduroam)

Once your server is ready, we can enable remote access. Please note that this step is optional. However, it is very convenient to enable remote access if you don't have spare peripherals for your Raspberry Pi, or you don't want to be working with the GUI of the virtual machine.

Secure Shell (SSH) is the best protocol to remotely access the edge server, it is already installed in Raspbian but it needs to be enabled. Open a terminal in your edge server and execute the following:

```
sudo systemctl enable ssh.servicesudo
systemctl start ssh.service
```

Now you can remotely access to your edge server executing the following in a terminal of your GNU/Linux or MacOSX computer:

```
ssh pi@<RPI_IP_ADDRESS>
```

The IP address of my (Lecturer) edge server is:

```
ssh pi@192.168.1.181
```

If you are using Windows, you can remotely access installing PuTTY or using your favourite SSH client.

Task 4: Connecting Arduino board to edge server via serial communication

Refer to the video file "Task 4 Demo.mp4" in the resources folder for video instructions to do this task.

Once our edge server is ready, we will connect it to our sensors (Arduino board) via serial communication. However, first we will see how to send and read data from serial port. Follow th steps below to connect Arduino board to edge server via serial communication.

1. For more information about serial communication, please check the following link: https://learn.sparkfun.com/tutorials/serial-communication/all

2. Connect the Arduino board to your computer running the virtual Raspberry Pi desktop (or your Raspberry Pi) using a serial cable.

3. Run the serial communication program from Task 2 on Arduino IDE to print via serial port "IOT Programming is GREAT!". Set the baud rate to 9600.

4. Put a small delay to not saturate the serial communication.

5. When your program runs successfully on Arduino, you may check it on the serial monitor. It shoulddisplay: IOT Programming is GREAT!

6. Check the communication port your Arduino is using. Typically /dev/ttyACM0 in GNU/Linux; and COM in Windows.

7. If using a Raspbian virtual machine, go to the machine settings in VirtualBox to configure the Serial Ports.

   Settings -> Serial Ports -> Port 1:

   Enable Serial Port

   Port Number: COM1 (in Raspberry Pi OS it will be /dev/ttyS0)

   Port Mode: Host Device

   Path/Address: YOUR_ARDUINO_COMMUNICATION_PORT (Typically /dev/ttyACM0 in GNU/Linux; and COMX in Windows, X is a natural number.)

8. You can test if the serial communication is working typing the following in your edge server (WARNING: It seems this did not work for everyone, you can jump this step):

   ```
   cat /dev/ttyS0
   ```

   Ctrl + c to close it.

9. Programming the edge server. Write a small code in Python (or C) in your Raspberry Pi OS to print the information received from the Arduino board via serial communication. You can do that in different ways:

   a) using the GUI of Raspberry Desktop: Applications menu -> Programming -> Thonny

Python IDE.

b) or like a pro, via SSH using your favourite editor (nano, vim), and executing the program: python your_serial_code.py

10. Once your code is working (it should display: IOT Programming is GREAT!), show it to your tutor!

Resources

Arduino

- Arduino Language Reference: https://www.arduino.cc/reference/en/
- Arduino Foundations: https://www.arduino.cc/en/Tutorial/Foundations
- Arduino Built-In Examples: https://www.arduino.cc/en/Tutorial/BuiltInExamples

Raspberry Pi

- https://www.raspberrypi.org/

Secure Shell

- https://www.ssh.com/ssh/
- SSH Client for Windows: https://www.ssh.com/ssh/putty/download

GNU/Linux

- Linux Journey is a site dedicated to making learning Linux fun and easy. https://linuxjourney.com/
- Introduction to Linux: A Hands on Guide. https://tldp.org/LDP/intro-linux/intro-linux.pdf
- Introduction to Linux (LFS101), the Linux Foundation training course: https://training.linuxfoundation.org/training/introduction-to-linux/