Lab02 - Arduino and I2C

MTRN3100 - UNSW School of Mechanical and Manufacturing Engineering

Lab Prerequisites

Install Arduino through the link: here

Once installed use the Arduino Library manager to install the following:

- VL6180X by PoloLU
- MPU6050_light by rfetick
- Adafruit SSD1306 by Adafruit
- Adafruit GFX Library by Adafruit

NOTE: If you get a serial port access denied error while trying to upload your code, close all other windows and the serial monitor before uploading. The serial monitor can potentially block the IDE writing to the port.

Task 1 - Digital Output

The "blinkDemo.ino" performs a simple blink program that turns the LED on pin 13 on and off. Connect an LED and Resistor (330 Ohm) to pin 9. Modify the code such that it blinks that LED.

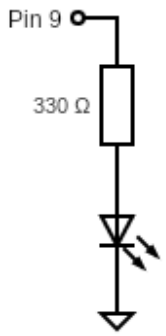

Figure 1: Circuit for task 1.

Questions

- **Question 1:** What is the voltage of an Arduino pin when set to "High"? (Use a multimeter to check)

Task 2 - Digital Input

Connect the nominally open button between pin 2 and GND on the Arduino. Modify your code from Task 1 such that pressing the button turns on the LED, otherwise, the LED is off. The reference documentation for the digital input command can be found: here.

The atmega328p (Arduino Nano) chip has internal pullup resistors that can be enabled, to do this specify "pinMode(inPin, INPUT_PULLUP);". If using this, no external pull-up/down resistor is required. The input pullup reference sheet can be found: here

Questions

- **Question 2:** Why do we need to specify INPUT_PULLUP, what would happen if it was not included? Hint: Consider the floating pin.
- **Question 3:** The atmega328p chip only has pull-up resistors and no pull-down resistors. Why is this the case?
- **Question 4:** If using a microcontroller with no internal pullup resistors, what may you need to do?

Task 3 - Serial Monitor

Open and run the provided "serialDemo.ino" program. Modify it such that it prints a counter that increments every loop. The output should look like:

Counter: 0

Counter: 1

Counter: 2

...

Questions

- **Question 5:** What is a baud rate? What happens if they are mismatched?
- **Question 6:** What does RX and TX mean?
- **Question 7:** If using two serial devices, how should the RX and TX pins be connected?

Task 4 - Read I2C address

Wire up one of the VL6180X lidars. The product information can be found: here. Run the "i2c_scanner.ino" and record the base address of the lidar. **NOTE: You may need to change the baud rate.**

Questions

- **Question 8:** What does SCL and SDA stand for?
- **Question 9:** What is the address of the Lidar?

Task 5 - Single Lidar

Create a new file that reads the lidar distance at a frequency of 10Hz and outputs the distance to the serial monitor. You need to use the documentation and library to find the necessary code.

Documentation and examples for the VL6180X lidar can be found: here

Distance: 22 mm

Distance: 22 mm

Distance: 22 mm

...

Questions

- **Question 10:** What is the default minimum and maximum range of the Lidar?
- **Question 11:** If you attach another lidar sensor what problem are you likely to encounter?

Task 6 - Double Lidar

Connect a second VL6180X lidar. Both sensors will have the same address, as such one will have to be changed. This can be done using the enable pin GPIO0/CE and the "setAddress()" function. Wire one lidar enable pin to A0, the other enable pin to A1. Upload and run the code supplied in "twoLidar.ino".

Questions

- **Question 12:** What is the maximum number of addresses and as such devices that can be on an I2C bus?

Task 7 - IMU

Create a new file that reads the yaw angle of the IMU. You need to use the documentation and examples to find the necessary code. Print the yaw value in degrees to the serial monitor.

Documentation and examples for the MPU6050_light library can be found: here

Yaw: 1.0 deg

Yaw: 1.2 deg

Yaw: 1.4 deg

...

Questions

- **Question 13:** What sensors are included in the MPU6050 package?

Extension Task 1 - Display

Use the OLED screen to display either a lidar distance or IMU yaw, rather than the serial monitor. Run and then modify the provided "ssd1306_128x64_i2c.ino" script. Look at the function void testdrawstyles(void) on line 315.

NOTE: the OLED screens used in this course have a default address 0x3C. Change line 35 in the sample code to use this address.

Documentation and examples for the OLED screen can be found: here

Extension Task 2 - Moving Average Filter

Using a fixed-length array of an arbitrary window size, implement a moving average filter to process the IMU yaw reading. You may choose to make this a class.