**Outline**

Learn the Arduino IDE and basic Arduino programming by implementing the basic blink program and modifying it to blink external LEDs.

**Objectives**

* Use constants and variables,

**Prerequisites**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prerequisite Module(s)** | **Level** | **Student Initial** | **Teacher Initial** | **Date** |
| None |  |  |  |  |

**Materials**

* Arduino Development Environment (IDE)
* Arduino proto board
* Documentation – See Appendix

**Level 0: Code & Run Basic Program**

1. Read the documentation to become familiar with the Arduino IDE.
2. Create a project on the network drive for this module.
3. Code and run the program to make the onboard LED (port 13) blink.

**HC-SR04: Ultrasonic Rangefinder**

Description:

Online Resources:

Sample Program:

#include <NewPing.h>

#define TRIGGER\_PIN 12

#define ECHO\_PIN 11

#define MAX\_DISTANCE 200

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

void setup() {

Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.

}

void loop() {

delay(50);

unsigned int uS = sonar.ping();

Serial.print("Ping: ");

Serial.print(uS / US\_ROUNDTRIP\_CM);

Serial.println("cm");

}

### END ###

**Part 1: Sensor Example**

1. List the members in your group:
2. List your assigned sensor type:
3. How does your sensor work? What does it do?
4. What did the example Arduino program do?

**Part 2: Background Thinking**

1. List three current applications of your sensor (Think about products in your home / school/ businesses):
2. Design an application using your sensor. It must be different than the current applications listed above. Creativity will be rewarded.  
   1. Describe your application.

* 1. List any objects (besides the sensor and Arduino board) involved in your application
  2. List any dimensions or timeouts (e.g. distances, movement times, etc.) for your application.

**Part 3: Designing Your Application**

1. List and explain the Inputs for your application. (e.g. Input sensors / devices, Input actions, Commands typed to the Serial Monitor, etc.):
2. List and explain the Outputs for your application. (e.g. Output devices, Output actions, Messages printed to the Serial Monitor, etc.)
3. Complete the following table for all important input / output actions (program logic).

|  |  |  |
| --- | --- | --- |
| **Input** | **Action (Program Logic)** | **Output** |
| (e.g. Red Button) | While the button is pressed, light up the LED | (e.g. Red LED) |
| (e.g. Red Button) | If the button is released then turn off the LED | (e.g. Red LED) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Part 4: Designing Your Program**

1. My assigned team role / responsibility is the following (check all that apply)
   * Wiring & Situation Diagram Specialist
   * Initialization “setup()” Code Specialist
   * Main “loop()” Code Specialist #1
   * Main “loop()” Code Specialist #2
2. Create a Wiring Diagram showing how your sensor and other input / output devices will be connected to your Arduino board.
3. Create a Situation Diagram showing how your sensor will work in the application (e.g. Show the room / location where your sensor is located, Show the people moving to trigger the sensor, etc.).
4. Create a list of the things that need to be initialized in "setup()" code block.
   1. Global constants for Arduino I/O Pin Numbers for sensor, LEDs, etc.
   2. Initialization of Sensor Functions, LED state, etc.
   3. Initialization of serial monitor, etc.
5. Create a list of variables to store integer and string information that the program will need when it operates. These variables may be used in both the “setup()” and “loop()” blocks.
6. Use the actions table in Part 3 to list the exact order sequence the main actions have to occur in the “loop()” block.
7. Use the actions table in Part 3 to create “IF … THEN…” condition statements for all actions that depend on external factors.
8. Use the actions table in Part 3 to create “FOR …” loop statements for all actions that have to repeat for a number of times.