Introduction to Arduino and Raspberry Pi

Presented by SEAS Computing Facility

March 24, 2018
Components
Jumper Cables
Breadboard

Diagram from Tweaking4All
Push Button

- All four pins are **connected** when pressed on
- If off, the **left** and **right** are separated
Light Emitting Diode

- A **diode** is a component that only allows flow of current in one direction.
- A **light emitting diode (LED)** emits light when current passes in the correct direction.
Circuit Basics

- Ohm’s Law: \( V = IR \)
  - \( V \): Voltage (volts)
  - \( I \): Current (amperes)
  - \( R \): Resistance (ohms)

- LEDs have a maximum current

- Ohm’s Law (rewritten): \( I = \frac{V}{R} \)
  - To keep current (I) low, resistance (R) must be high enough

---

**Table: Absolute Maximum Rating**

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Absolute Maximum Rating</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Current</td>
<td>If</td>
<td>20</td>
<td>mA</td>
</tr>
<tr>
<td>Peak Forward Current</td>
<td>If'</td>
<td>120</td>
<td>mA</td>
</tr>
<tr>
<td>Reverse Voltage</td>
<td>Vr</td>
<td>5</td>
<td>V</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>Pd</td>
<td>85</td>
<td>W</td>
</tr>
<tr>
<td>Operation Temperature</td>
<td>Topr</td>
<td>-35°C to 80°C</td>
<td></td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>Tstg</td>
<td>-40°C to 80°C</td>
<td></td>
</tr>
<tr>
<td>Lead Soldering Temperature</td>
<td>Tsol</td>
<td>Max. 200°C for 3sec Max.</td>
<td></td>
</tr>
</tbody>
</table>

*If Conditions: Pulse Width 10μsec duty ≤ 1/10
Tsol Conditions: 4mm from the base of the epoxy bulb
Raspberry Pi
What is a Raspberry Pi?

- Single-board computer
- Developed in the UK
- Several models
- Inexpensive ($5 for cheapest model, the Raspberry Pi Zero)
- Can be used with a computer monitor, keyboard, and mouse
What is Raspbian?

● Operating system optimized for the Raspberry Pi
● Based on the Linux kernel
● Can be used like a desktop computer or through the terminal
Raspberry Pi 3 Model B

- Dimensions: 85.6mm x 56mm x 21mm
- 4 x USB 2 Ports
- 10/100 LAN Port
- 3.5mm 4-pole Composite Video and Audio Output Jack
- CSI Camera Port
- Full Size HDMI Video Output
- Broadcom BCM2837 64bit Quad Core CPU at 1.2GHz, 1GB RAM
- On Board Bluetooth 4.1 Wi-Fi
- MicroSD Card Slot
- DSI Display Port
- 40 Pin Extended GPIO
- Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps
Connecting to the Internet

- **Wired Connection**: The SEASCF Raspberry Pis can instantly connect to the GW network from the SEH Studio Labs using an ethernet cable. This is the easiest and fastest option.
- **GWWireless**: Raspberry Pis cannot connect to GWWireless.
- **eduroam**: Raspberry Pis can be connected to eduroam by modifying two configuration files and running commands.
  - Instructions: [seascf.seas.gwu.edu/eduroam-connection](http://seascf.seas.gwu.edu/eduroam-connection)
    - It may be necessary to run `/etc/init.d/networking stop` before running `/etc/init.d/networking start`
Creating a Basic Python Program

1. Open Terminal
2. Type `nano helloworld.py` and press ENTER to open a new file in the nano text editor
3. Type `print("Hello, World!")`
4. Use CTRL + O and ENTER to save
5. Exit with CTRL + X
6. Type `python helloworld.py` and press ENTER to run the program
7. Hello, World! should appear
Connecting to a Breadboard
Connecting to a Breadboard
Connecting to a Breadboard
Building the LED Circuit

- Two jumper cables
- LED
- 220 Ω resistor
Building the LED Circuit

- Add a resistor to the breadboard
- Connect a wire from the red power rail to one end of a resistor
Building the LED Circuit

- Connect the anode (long end) of the LED to the resistor
Building the LED Circuit

- Connect a wire from the cathode (short end) of the LED to the blue ground rail of the breadboard
- The LED should glow!
Blinking an LED

1. Connect the cable from the resistor to pin 21 of the breakout board
2. Open Terminal
3. Type `nano blinky.py` and press ENTER to open a new file in the nano text editor
4. Type the code
5. Use CTRL + O and ENTER to save
6. Exit with CTRL + X
7. Type `python helloworld.py` and press ENTER to run the program

---

**Code:**
```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(21, GPIO.OUT)

for i in range(0,100):
    GPIO.output(21, i % 2)
    time.sleep(0.25)

GPIO.cleanup(21)
```
Arduino
What is an Arduino?

- Single-board microcontroller
- Originated in Italy
- Many variations from different makers
- Cannot be directly connected to a monitor, keyboard, mouse, etc.
- Does not normally have an operating system
Arduino Uno

- Power In
  - Barrel Jack
  - USB

- Power Out
  - (3.3v and 5v)

- Ground
  - Power In

- Analog In

- Digital In/Out
  - PWM (3, 5, 6, 9, 10, 11)

- RX/TX

- ARef

- Ground

- Reset
Running a Basic Arduino Program

1. Open the Arduino Desktop IDE (install required)
   a. There is also an Arduino Web IDE (account required)
2. Connect the Arduino to the computer using a USB cable
3. Open example sketch from File > Examples > 01.Basics > Blink
4. Select the board type from Tools > Board
5. Select the port with the Arduino from Tools > Port
6. Click the upload button
7. The built-in LED near pin 13 should start to blink

Programs will stay on the Arduino until overridden by another program.

Programs start whenever the Arduino is powered on or reset.
Running a Basic Arduino Program

Code:

```cpp
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```
Useful Arduino Functions

- `pinMode(pin, mode)` - Sets the pin to be INPUT or OUTPUT
- `digitalRead(pin)` - Returns HIGH or LOW depending on the voltage of the specified pin
- `analogRead(pin)` - Returns a value from 0 to 1023 representing the voltage of the specified pin between 0 and 5 V
- `analogWrite(pin, value)` - Writes a value between 0 and 255 to the pin
- `digitalWrite(pin, value)` - Writes HIGH or LOW to the pin
- `Serial.begin(speed)` - Configures the serial output for the specified speed (9600 is typical)
- `Serial.write(val)` - Writes a value or string to the Serial monitor
More Components
Raspberry Pi Camera Module V2

- 1080p HD video at 30 frames/second
- 720p HD video at 60 frames/second
- 8 Megapixels for still photos (3280 x 2464)
- Fixed Focus Lens
- Connected to Raspberry Pi with 15-pin ribbon cable

https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS
PIR Motion Sensor Detector Module

- **PIR**: Passive Infrared
  - Senses infrared radiation from objects
- Range is adjustable up to 7 meters
- Viewing area is approximately a 120° cone

PIR Motion Sensor Detector Module

- **Time Delay Adjust**
  - Clockwise increases delay
- **Sensitivity Adjust**
  - Clockwise decreases range
- **Pins**
  - **Power**: Should be between 5 and 20 V input
  - **Ground**: Should be connected to ground
  - **Output**: Will be 3.3 V if activated, 0 if not

Diagram from [Henry’s Bench](#)
Sense HAT for Raspberry Pi

- Includes:
  - 8 x 8 RGB LED matrix
  - Five-button joystick
  - Gyroscope
  - Accelerometer
  - Magnetometer
  - Thermometer
  - Barometric pressure
  - Humidity

- Has been used on the International Space Station
Sense HAT Basics

Install the Sense HAT package

- sudo apt-get install sense-hat
- Ensure that you **unplug** your Pi
- Attach your Sense HAT
Setting up the code and sending text to the HAT

- Create a Python file: `nano helloWorld.py`
- Create Sense HAT object:
  ```python
  from sense_hat import SenseHat
  sense = SenseHat()
  ```
- Have text scroll across the Sense HAT
  ```python
  sense.show_message("Hello world")
  ```
Further Information
# Raspberry Pi vs. Arduino

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Arduino</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Allows graphical user interface</td>
<td>● Low power consumption</td>
</tr>
<tr>
<td>● Can be directly connected to Internet</td>
<td>● Can directly read analog inputs</td>
</tr>
<tr>
<td>● More powerful and more memory</td>
<td>● Requires less hardware (monitor, mouse, etc.) to get started</td>
</tr>
<tr>
<td>● Can be used with more programming languages</td>
<td>● No operating system needs to be installed</td>
</tr>
</tbody>
</table>
Link to These Slides

seascf.seas.gwu.edu/workshops