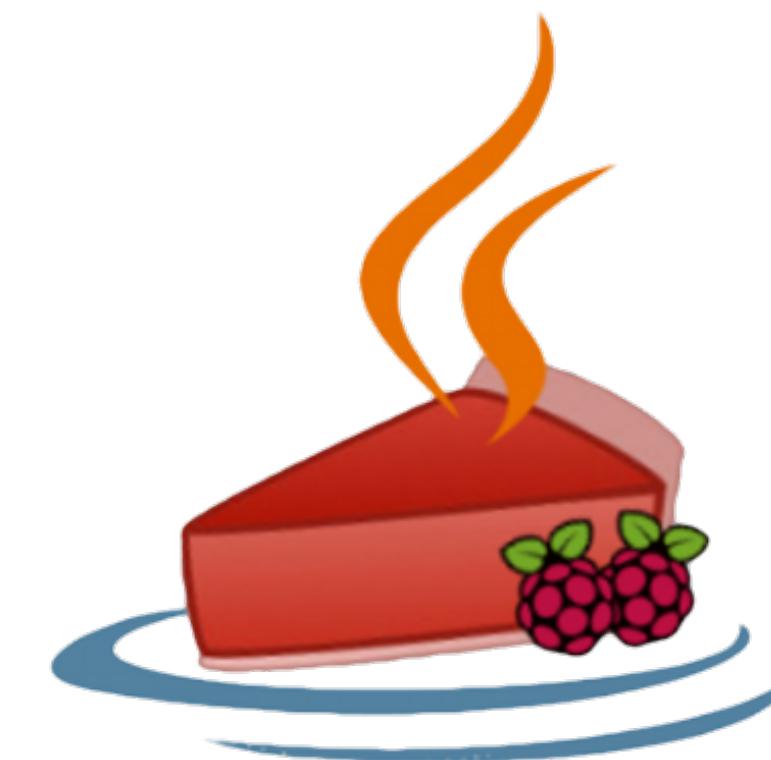


Raspberry Pi with Java 8 + Pi4J

Robert Savage
Software Architect, MTS
Harman International





What is Pi4J?

Pi4J is an open-source project providing a library for Java programmers to interact with the low-level I/O capabilities on the Raspberry Pi platform.

- Open Source Project
- Low Level I/O Library
- Object-Oriented API
- Event Based
- Java & C (JNI + Native)



www.pi4j.com

Pi4J Supported I/O

Digital Interfaces

- **GPIO**

(General Purpose Input/Output)

Data Interfaces

- **UART, SERIAL**
- **SPI**
- **I²C**

(Universal Asynchronous Receiver/Transmitter)

(Serial Peripheral Interface)

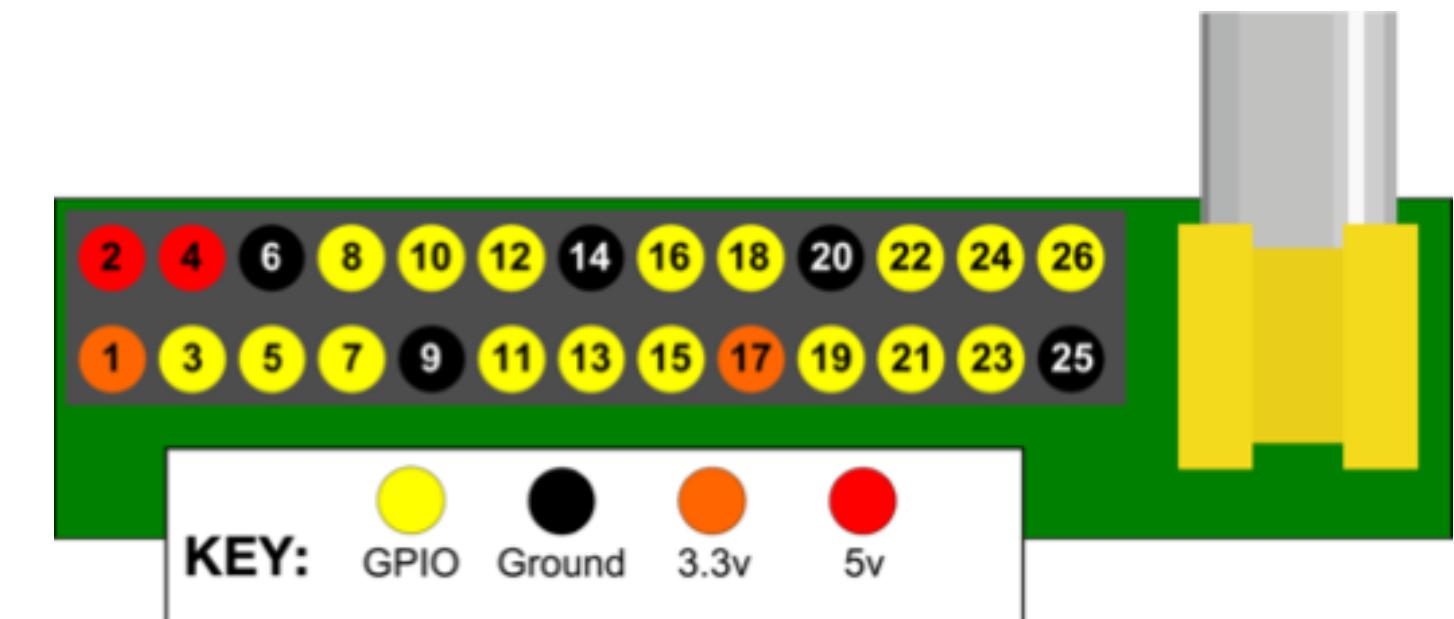
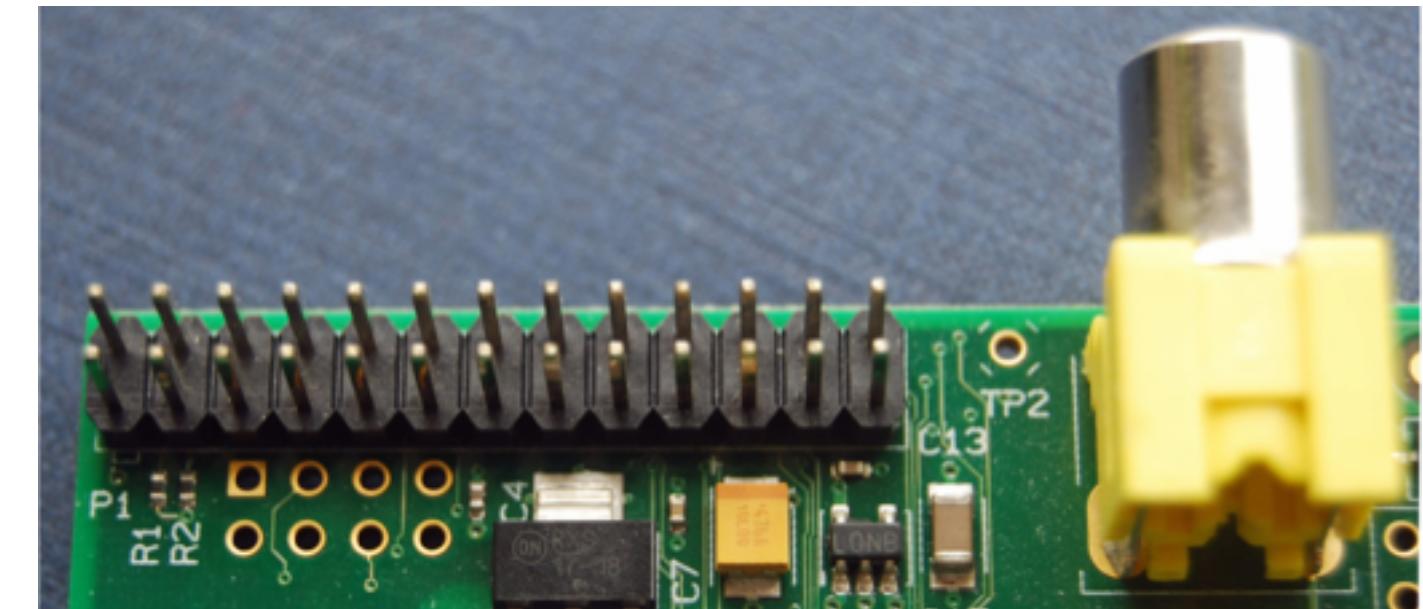
(Inter-Integrated Circuit)

Analog Interfaces*



GPIO

- **Input or Output**
- **Digital States**
 - HIGH = +3.3 VDC
 - LOW = 0 VDC
- **RPi Models**
 - A & B = ~21 GPIO
 - B+ = 28 GPIO
 - Compute Module = 46 GPIO





Pi4J GPIO Pin Addressing

| Raspberry Pi P1 Header | | | |
|------------------------|---------------|----------------|-------|
| PIN # | NAME | NAME | PIN # |
| | 3.3 VDC Power | 5.0 VDC Power | 1 |
| 8 | SDA0 (I2C) | DNC | 2 |
| 9 | SCL0 (I2C) | 0V (Ground) | 3 |
| 7 | GPIO 7 | TxD 15 | 4 |
| | DNC | RxD 16 | 5 |
| 0 | GPIO 0 | GPIO1 1 | 6 |
| 2 | GPIO2 | DNC | 7 |
| 3 | GPIO3 | | 8 |
| | DNC | | 9 |
| 12 | MOSI | | 10 |
| 13 | MISO | GPIO6 6 | 11 |
| 14 | SCLK | CE0 10 | 12 |
| | DNC | CE1 11 | 13 |

<http://www.pi4j.com>

| Raspberry Pi P5 Header | | | |
|------------------------|---------------|------------------|-------|
| PIN # | NAME | NAME | PIN # |
| | 3.3 VDC Power | 5.0 VDC Power | 1 |
| 18 | GPIO18 | GPIO17 17 | 2 |
| 20 | GPIO20 | GPIO19 19 | 3 |
| | 0V (Ground) | 0V (Ground) | 4 |
| 8 | | | 5 |
| | | | 6 |
| | | | 7 |

<http://www.pi4j.com>

| Raspberry Pi J8 Header (Model B+) | | | |
|-----------------------------------|-------------------|---------------------|-------|
| GPIO# | NAME | NAME | GPIO# |
| | 3.3 VDC Power | 5.0 VDC Power | 1 |
| 8 | GPIO 8 SDA1 (I2C) | 5.0 VDC Power | 2 |
| 9 | GPIO 9 SCL1 (I2C) | Ground | 3 |
| 7 | GPIO 7 GPCLK0 | GPIO 15 TxD (RS232) | 4 |
| | Ground | GPIO 16 RxD (RS232) | 5 |
| 0 | GPIO 0 | GPIO 1 PCM_CLK/PWM0 | 6 |
| 2 | GPIO 2 | Ground | 7 |

| | | | |
|----|----------------------|----|-----------------------------|
| 13 | MISO (SPI) | 21 | GPIO 6 6 |
| 14 | GPIO 14 SCLK (SPI) | 23 | GPIO 10 CE0 (SPI) 10 |
| | Ground | 25 | GPIO 11 CE1 (SPI) 11 |
| | SDA0 (I2C ID EEPROM) | 27 | SCL0 (I2C ID EEPROM) |
| 21 | GPIO 21 GPCLK1 | 29 | Ground |
| 22 | GPIO 22 GPCLK2 | 31 | GPIO 26 PWM0 26 |
| 23 | GPIO 23 PWM1 | 33 | Ground |
| 24 | GPIO 24 PCM_FS/PWM1 | 35 | GPIO 27 27 |
| 25 | GPIO 25 | 37 | GPIO 28 PCM_DIN 28 |
| | Ground | 38 | GPIO 29 PCM_DOUT 29 |
| 39 | | 40 | |

<http://www.pi4j.com>

| RPi Compute Mod Dev Board - J5 Header | | |
|---------------------------------------|--------------------------|------|
| GPIO# | NAME | NAME |
| 0 | GPIO 0 | 1 |
| 1 | GPIO 1 | 2 |
| 2 | GPIO 2 (I2C) SDA1 [ALTO] | 3 |
| 3 | GPIO 3 (I2C) SCL1 [ALTO] | 4 |
| 4 | GPIO 4 GPCLK0 [ALTO] | 5 |
| 5 | GPIO 5 | 6 |
| 6 | GPIO 6 | 7 |
| 7 | GPIO 7 SPI0_CE1_N [ALTO] | 8 |
| 8 | GPIO 8 SPI0_CE0_N [ALTO] | 9 |
| 9 | GPIO 9 SPI0_MISO [ALTO] | 10 |
| 10 | GPIO 10 SPI0_MOSI [ALTO] | 11 |

| | | | |
|----|---------------------|----|---------------|
| 16 | GPIO 16 | 33 | Ground |
| 17 | GPIO 17 | 35 | 1.8 VDC Power |
| 18 | GPIO 18 PWM0 [ALT5] | 36 | Ground |
| 19 | GPIO 19 PWM1 [ALT5] | 38 | VG0 Power |
| 20 | GPIO 20 | 40 | Ground |
| 21 | GPIO 21 | 42 | 3.3 VDC Power |
| 22 | GPIO 22 | 44 | Ground |
| 23 | GPIO 23 | 46 | 1.8 VDC Power |
| 24 | GPIO 24 | 48 | Ground |
| 25 | GPIO 25 | 50 | VG0 Power |
| 26 | GPIO 26 | 52 | Ground |
| 27 | GPIO 27 RUN | 54 | 5.0 VDC Power |
| 59 | GPIO47_CTL_1V8 | 56 | Ground |
| | | 58 | |
| | | 60 | Ground |

<http://www.pi4j.com>

| RPi Compute Mod Dev Board - J6 Header | | |
|---------------------------------------|---------|------|
| GPIO# | NAME | NAME |
| 28 | GPIO 28 | 1 |
| 29 | GPIO 29 | 2 |
| 30 | GPIO 30 | 3 |
| 31 | GPIO 31 | 4 |
| 32 | GPIO 32 | 5 |
| 33 | GPIO 33 | 6 |
| 34 | GPIO 34 | 7 |
| 35 | GPIO 35 | 8 |
| 36 | GPIO 36 | 9 |
| 37 | GPIO 37 | 10 |
| 38 | GPIO 38 | 11 |
| | | 12 |
| | | 13 |
| | | 14 |
| | | 15 |
| | | 16 |
| | | 17 |
| | | 18 |
| | | 19 |
| | | 20 |
| | | 21 |
| | | 22 |
| | | 23 |
| | | 24 |
| | | 25 |
| | | 26 |
| | | 27 |
| | | 28 |
| | | 29 |
| | | 30 |
| | | 31 |
| | | 32 |
| | | 33 |
| | | 34 |
| | | 35 |
| | | 36 |
| | | 37 |
| | | 38 |
| | | 39 |
| | | 40 |
| | | 41 |
| | | 42 |
| | | 43 |
| | | 44 |
| | | 45 |
| | | 46 |
| | | 47 |
| | | 48 |
| | | 49 |
| | | 50 |
| | | 51 |
| | | 52 |
| | | 53 |
| | | 54 |
| | | 55 |
| | | 56 |
| | | 57 |
| | | 58 |
| | | 59 |
| | | 60 |

<http://www.pi4j.com>



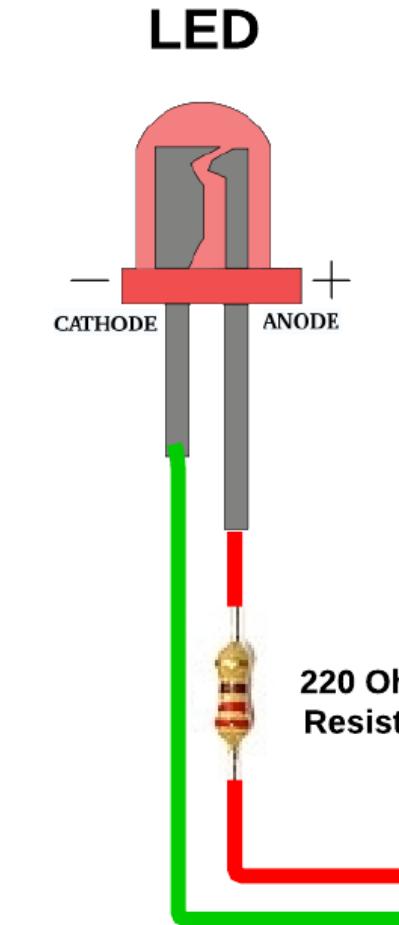
GPIO Outputs

- Control Things
(ON & OFF)





GPIO Output Circuit



| Raspberry Pi P1 Header | | | |
|------------------------|---------------|----|---------------|
| PIN # | NAME | | NAME PIN # |
| | 3.3 VDC Power | 1 | 5.0 VDC Power |
| 8 | SDA0 (I2C) | 3 | DNC |
| 9 | SCL0 (I2C) | 5 | 0V (Ground) |
| 7 | GPIO 7 | 7 | TxD 15 |
| | DNC | 9 | RxD 16 |
| 0 | GPIO 0 | 11 | GPIO1 1 |
| 2 | GPIO2 | 13 | DNC |
| 3 | GPIO3 | 15 | GPIO4 4 |
| | DNC | 17 | GPIO5 5 |
| 12 | MOSI | 19 | DNC |
| 13 | MISO | 21 | GPIO6 6 |
| 14 | SCLK | 23 | CE0 10 |
| | DNC | 25 | CE1 11 |

<http://www.pi4j.com>



GPIO Output Example

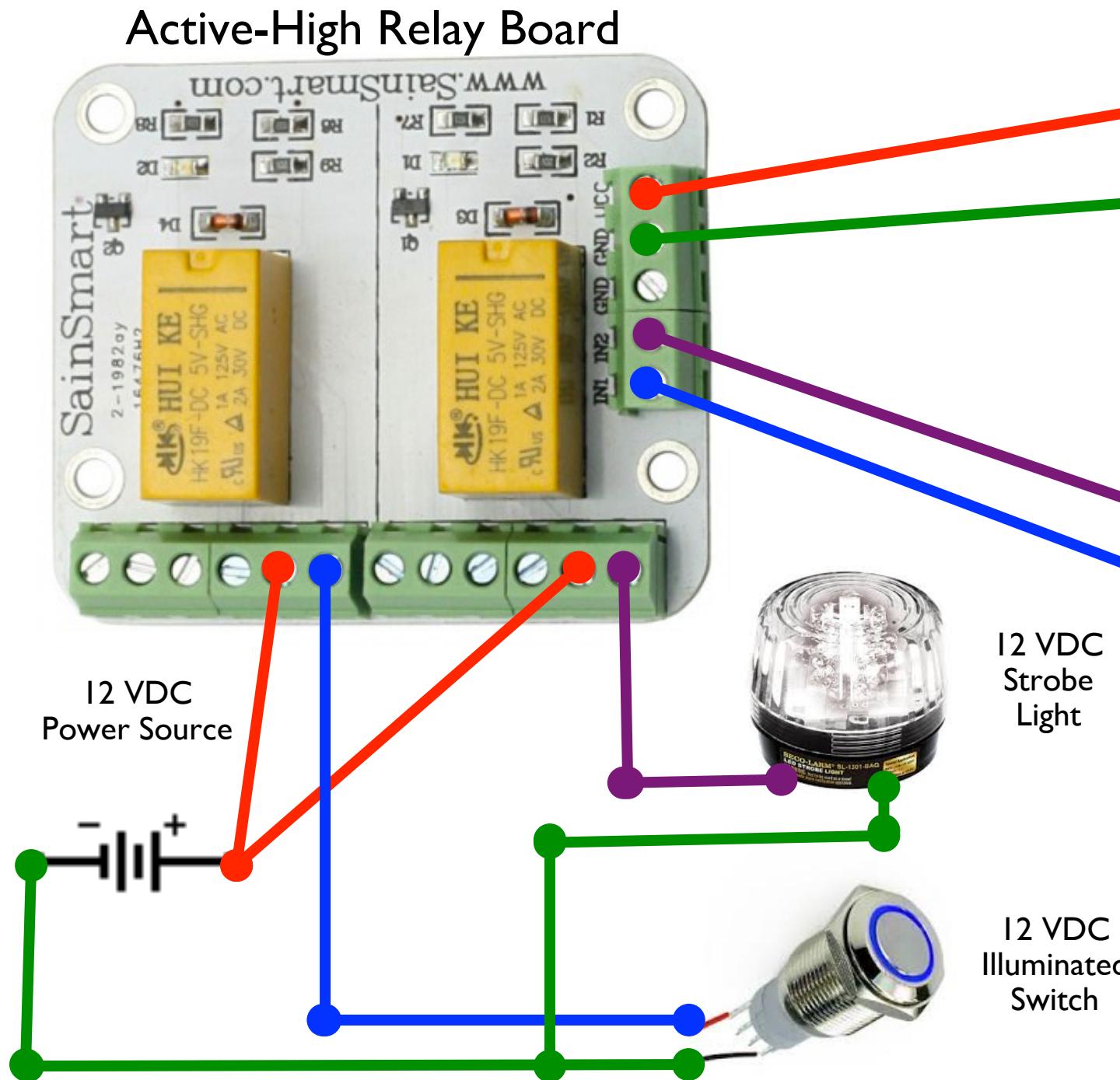
```
// create GPIO controller
final GpioController gpio = GpioFactory.getInstance();

// create GPIO output pin
final GpioPinDigitalOutput output =
    gpio.provisionDigitalOutputPin(
        RaspiPin.GPIO_12, PinState.LOW);

// control GPIO output pin
output.high();
output.low();
output.toggle();          // invert current state
output.pulse(1000);      // set state for a limited duration
```

TM
DEVOXX

GPIO Output Circuit



| Raspberry Pi P1 Header | |
|------------------------|---------------|
| PIN # | NAME |
| 1 | 3.3 VDC Power |
| 2 | 5.0 VDC Power |
| 4 | DNC |
| 6 | 0V (Ground) |
| 8 | SDA0 (I2C) |
| 9 | SCL0 (I2C) |
| 15 | TxD |
| 16 | RxD |
| 1 | GPIO1 |
| 2 | DNC |
| 4 | GPIO4 |
| 5 | GPIO5 |
| 6 | DNC |
| 10 | CE0 |
| 11 | CE1 |
| 12 | MOSI |
| 13 | MISO |
| 14 | SCLK |
| 17 | DNC |
| 20 | DNC |
| 21 | DNC |
| 22 | DNC |
| 23 | DNC |
| 25 | DNC |
| 26 | DNC |

<http://www.pi4j.com>

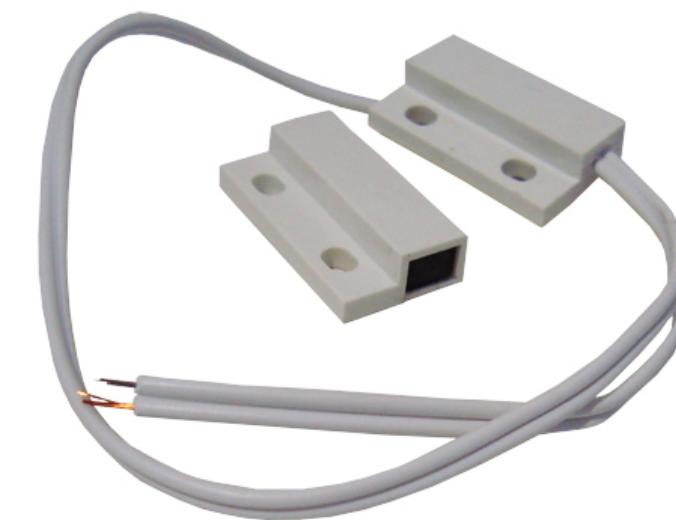


GPIO Output Demo

- Sample Code
- Live Demo

GPIO Inputs

- Monitor Things
 - ON & OFF
 - Open & Close



TM

GPIO Input Circuit

Momentary
Switch



NOTE:

GPIO pull-down or pull-up must be enabled to prevent the pin state from floating for the switch.

| Raspberry Pi P1 Header | | | |
|------------------------|---------------|----|---------------|
| PIN # | NAME | | NAME PIN # |
| | 3.3 VDC Power | | 5.0 VDC Power |
| 8 | SDA0 (I2C) | 3 | DNC |
| 9 | SCL0 (I2C) | 5 | 0V (Ground) |
| 7 | GPIO 7 | 7 | TxD 15 |
| | DNC | 9 | RxD 16 |
| 0 | GPIO 0 | 11 | GPIO1 1 |
| 2 | GPIO2 | 13 | DNC |
| 3 | GPIO3 | 15 | GPIO4 4 |
| | DNC | 17 | GPIO5 5 |
| 12 | MOSI | 19 | DNC |
| 13 | MISO | 21 | GPIO6 6 |
| 14 | SCLK | 23 | CE0 10 |
| | DNC | 25 | CE1 11 |

<http://www.pi4j.com>



GPIO Input Reference

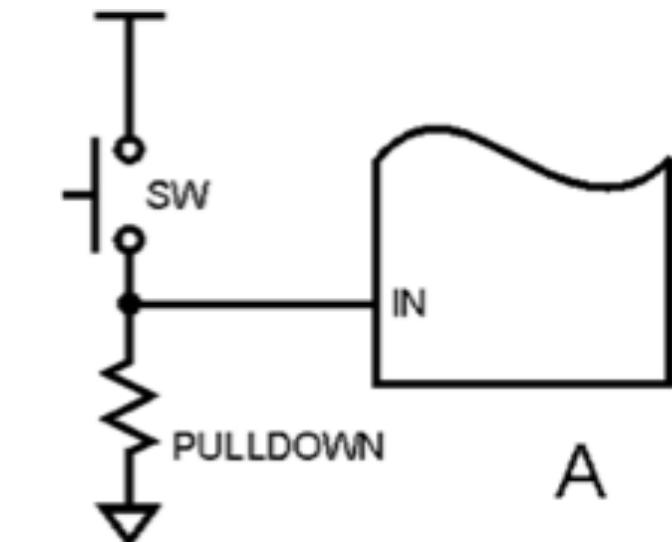
- GPIO inputs require a “reference” voltage.
- Without a reference, a GPIO pin can “float”.
- The Raspberry Pi includes internal **PULL-UP** and **PULL-DOWN** resistor settings that can be configured via Pi4J.



GPIO Input Reference

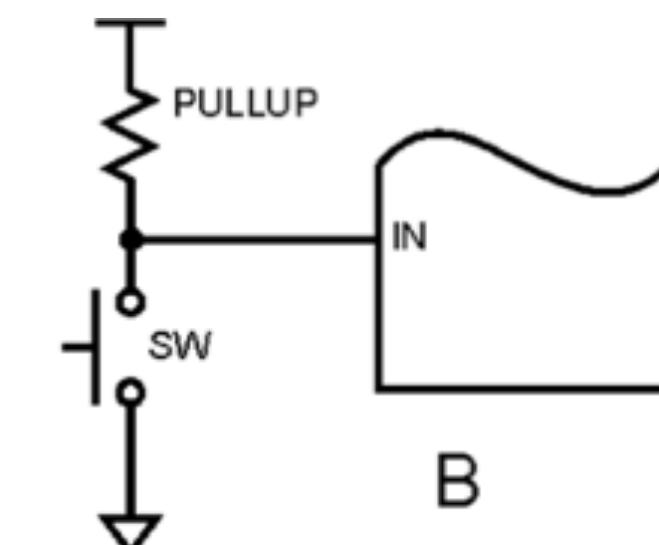
- **PULL-DOWN**

Resistance provides a reference (bias) to GROUND (0 VDC). If your circuit expects to provide +3.3VDC to signal the GPIO pin HIGH, then you need a PULL-DOWN reference.



- **PULL-UP**

Resistance provides a reference (bias) to +3.3 VDC. If your circuit expects to provide GROUND to signal the GPIO pin LOW, then you need a PULL-UP reference.

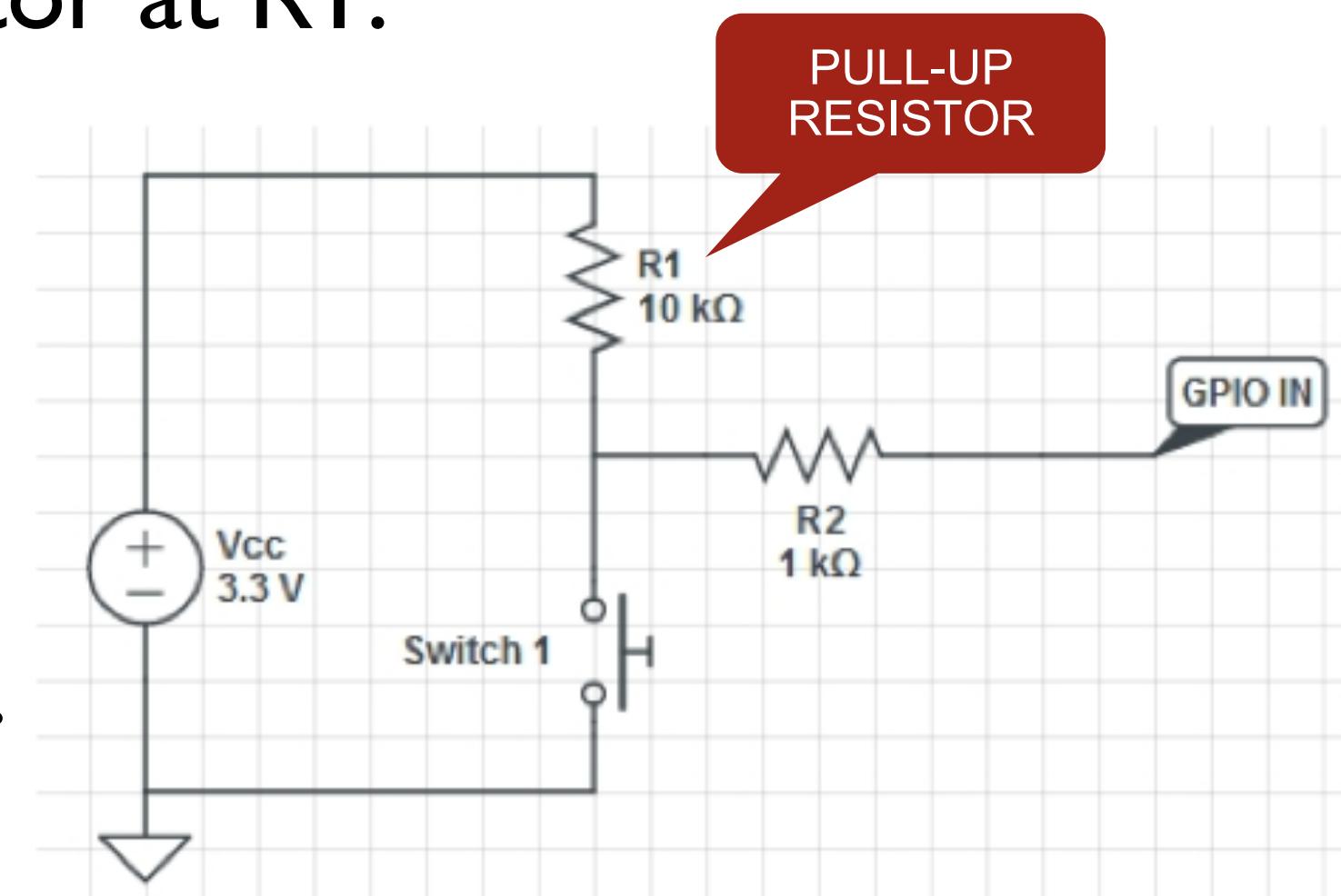




GPIO Input Reference

Alternatively, you can build the PULL-UP or PULL-DOWN reference in the hardware circuit. The circuit below demonstrates a PULL-UP resistor at R1.

This circuit signals the GPIO input pin to LOW when the switch closes the circuit and a path to GROUND is complete.





GPIO Input Example

```
// create GPIO controller
final GpioController gpio = GpioFactory.getInstance();

// create a GPIO input pin
final GpioPinDigitalInput input = gpio.provisionDigitalInputPin(
    RaspiPin.GPIO_02,
    PinPullResistance.PULL_DOWN);
```



GPIO Input Listener Example

```
// create event listener for GPIO input pin
input.addListener((GpioPinListenerDigital)
    (GpioPinDigitalStateChangeEvent event) -> {

    // set output state to match the input state
    output.setState(event.getState());
});
```

Java 8
Lambda



GPIO Input Demo

- Sample Code
- Live Demo



Pi4J Component API

- The component APIs provides an abstraction layer from the hardware I/O layer.
- This allows hardware design/circuitry to change with *less* impact to your implementation code.
- For example, a RELAY could be controlled from GPIO, RS232, SPI, or I2C. Your program defines the RELAY impl up front based on the hardware interface, but the rest of your program logic works against the RELAY component interface and not the direct hardware /communication IO interfaces.



Component Interfaces

- Keypad
- Light / LED
- Dimmable Light
- LCD
- Power Controller
- Relay
- Momentary Switch
- Toggle Switch
- Analog Sensor
- Distance Sensor
- Motion Sensor
- Temperature Sensor



GPIO Components Example

```
// create LED component
final Light light = new GpioLightComponent(output);

// usage example
light.on(); (or) light.off();

// create momentary switch component
final MomentarySwitch ms = new GpioMomentarySwitchComponent(
    input,
    PinState.LOW, // "OFF" pin state
    PinState.HIGH); // "ON" pin state
```



Component Demo

- Sample Code
- Live Demo

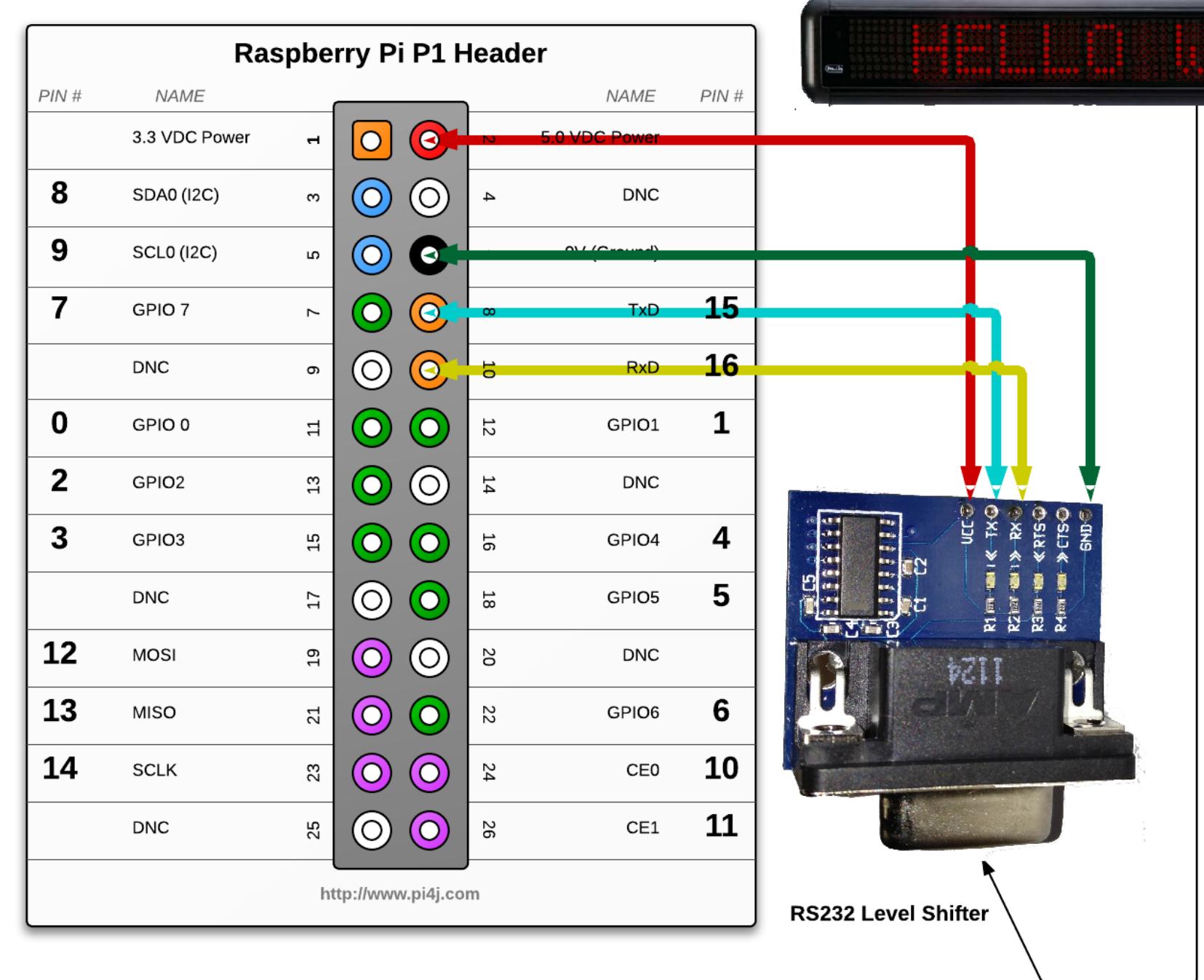


UART / Serial / RS-232

- The Raspberry Pi supports one on-board UART for serial communication.
- Pi4J supports basic serial communication.
- To use RS232 serial communication a level-shifter is required to convert between the TTL voltage levels (+3.3 VDC) and those required for RS232 communication.

TM
DEVOXX

RS-232 Level Shifter



@savageautomate

#DV14 #pi4j



USB to RS-232 Adapter

In addition to the on-board UART, you can also use a USB to RS232 adapter connected to the Raspberry Pi to provide RS232 serial communication.





UART / RS-232 Example

```
// create an instance of the serial communications class
final Serial serial = SerialFactory.createInstance();

// open the default serial port provided on the P1 header
// (this is where our LED reader is connected)
serial.open(Serial.DEFAULT_COM_PORT, 2400);

// send "Hello World" message to sign
serial.writeln("<ID01><PA><CL>Hello World! -- ");
serial.writeln("<ID01><RPA>");
```



UART / RS-232 Listener Example

Java 8
Lambda

```
// create and register the serial data listener
serial.addListener((SerialDataListener) (SerialDataEvent event) -> {

    // print out the data received to the console
    System.out.println(event.getData());

});
```



UART / RS-232 Demo

- Sample Code
- Live Demo



Putting It All Together

Lets create a real-world demo using a Raspberry Pi, Java, Pi4J, GPIO Inputs & Outputs, and RS232 (Serial) devices to do something *useful*.



Demo Project Goal

- Create a sophisticated model rocket launching platform.





Demo Project Requirements

- Implement a safety key switch to arm the rocket and protect from accidental launch ignition.
- Implement a launch activation button to initiate a launch sequence.





Demo Project Requirements

- Implement an audible safety alarm to alert spectators of launch ready conditions.
- Implement a visible alert device to notify spectators of launch ready conditions.





Demo Project Requirements

- Implement an abort button to abort the launch sequence.
- Implement a safety IR beam detector to abort launch if a person approaches the launch pad.





Demo Project Requirements

- Implement an LED message board to show spectators the countdown and phases of the launch.



- Implement a countdown timer for the launch sequence.

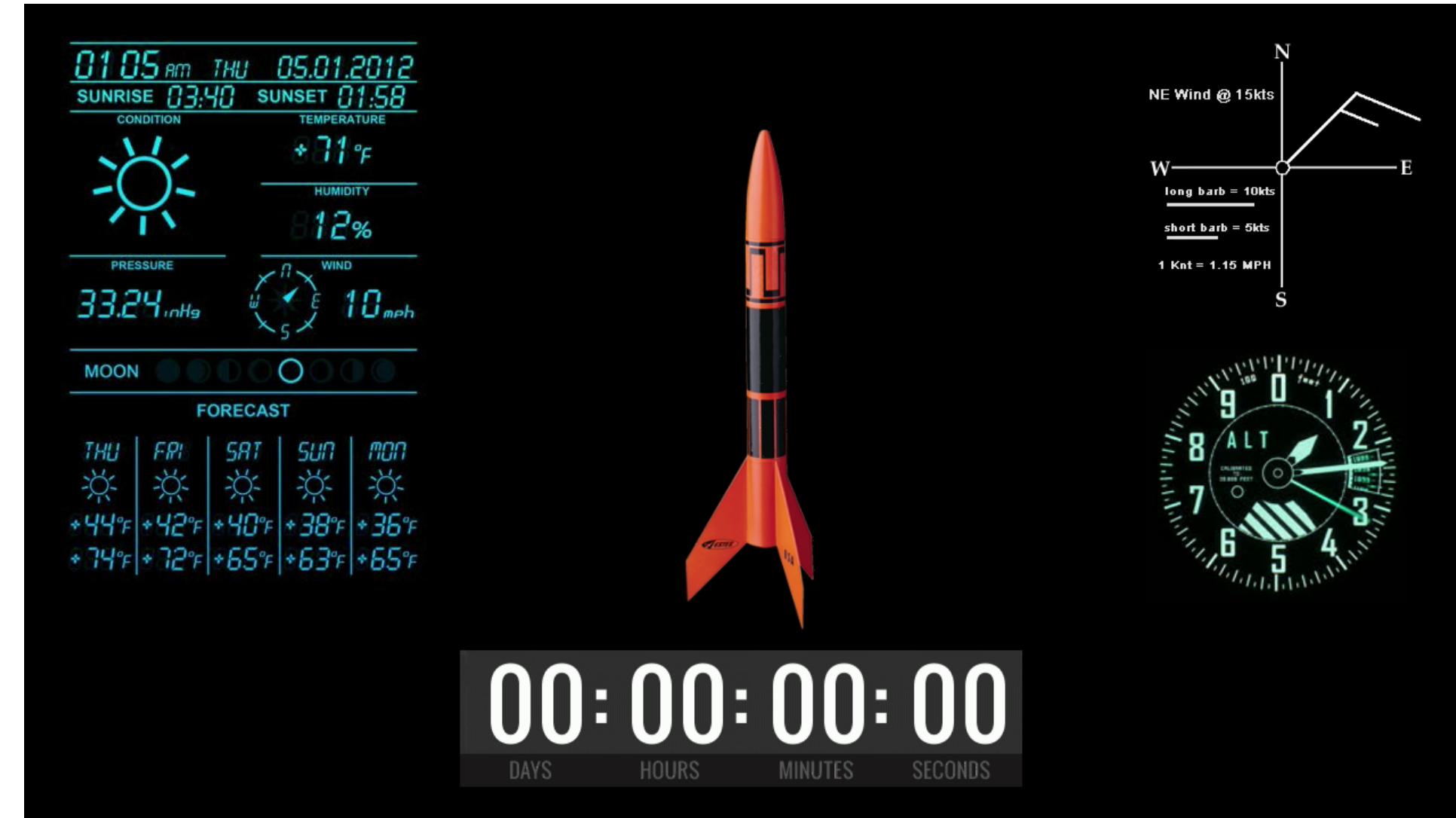


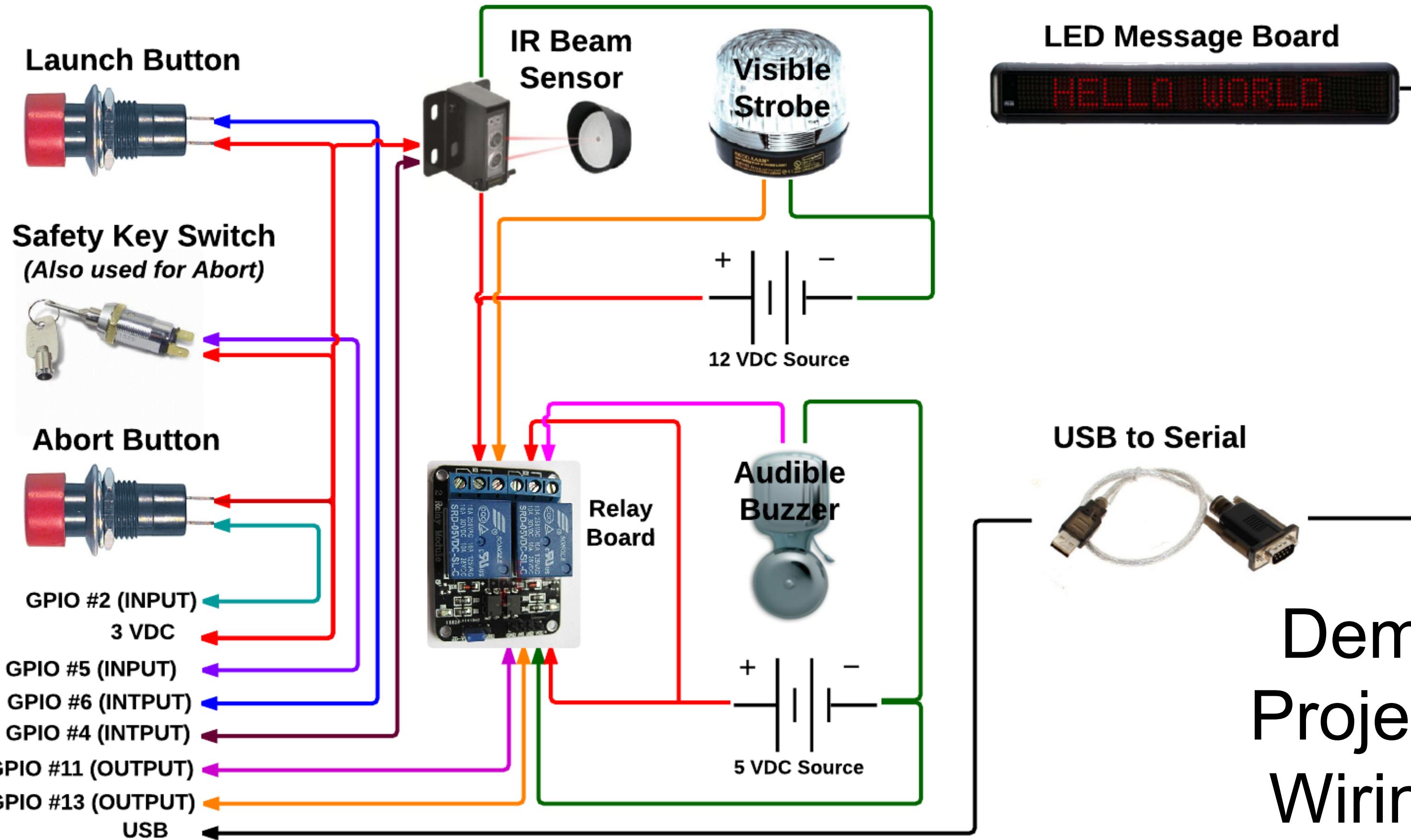
TM



Demo Project Requirements

- Implement an on-screen console/dashboard







Demo Project

- Sample Code
- Live Demo



Savage Home Automation Blog



savagehomeautomation.com



[@savageautomate](https://twitter.com/savageautomate)



The Pi4J Project



pi4j.com



[@pi4j](https://twitter.com/pi4j)