



BUILD YOUR OWN GAME CONTROLLER

Josef Spjut
SIGGRAPH 2015



AGENDA

- 5 minutes: Intro
- 10 Minutes: Arduino LED blink example
- 10 Minutes: Arduino button example
- 20 Minutes: Arduino Joystick example
- 10 Minutes: Place buttons and joysticks on controller base
- 10 Minutes: Wiring
- 10 Minutes: Bring it all together (integrate software and hardware)
- 15 Minutes: Wrap up and testing.

ASSIGNMENT

- ▶ Build a game controller from the included parts
- ▶ Program the game controller
 - ▶ Read inputs
 - ▶ Send USB Mouse + Keyboard to PC

ACKNOWLEDGEMENTS

- ▶ Developed at Harvey Mudd College
- ▶ Wooden Controller Base
 - ▶ Richard Piersall (Harvey Mudd College '16)
 - ▶ Kirklann Lau (Harvey Mudd College '16)
- ▶ Arduino Platform
 - ▶ arduino.cc

WHY BUILD OUR OWN GAME CONTROLLER?

Learning

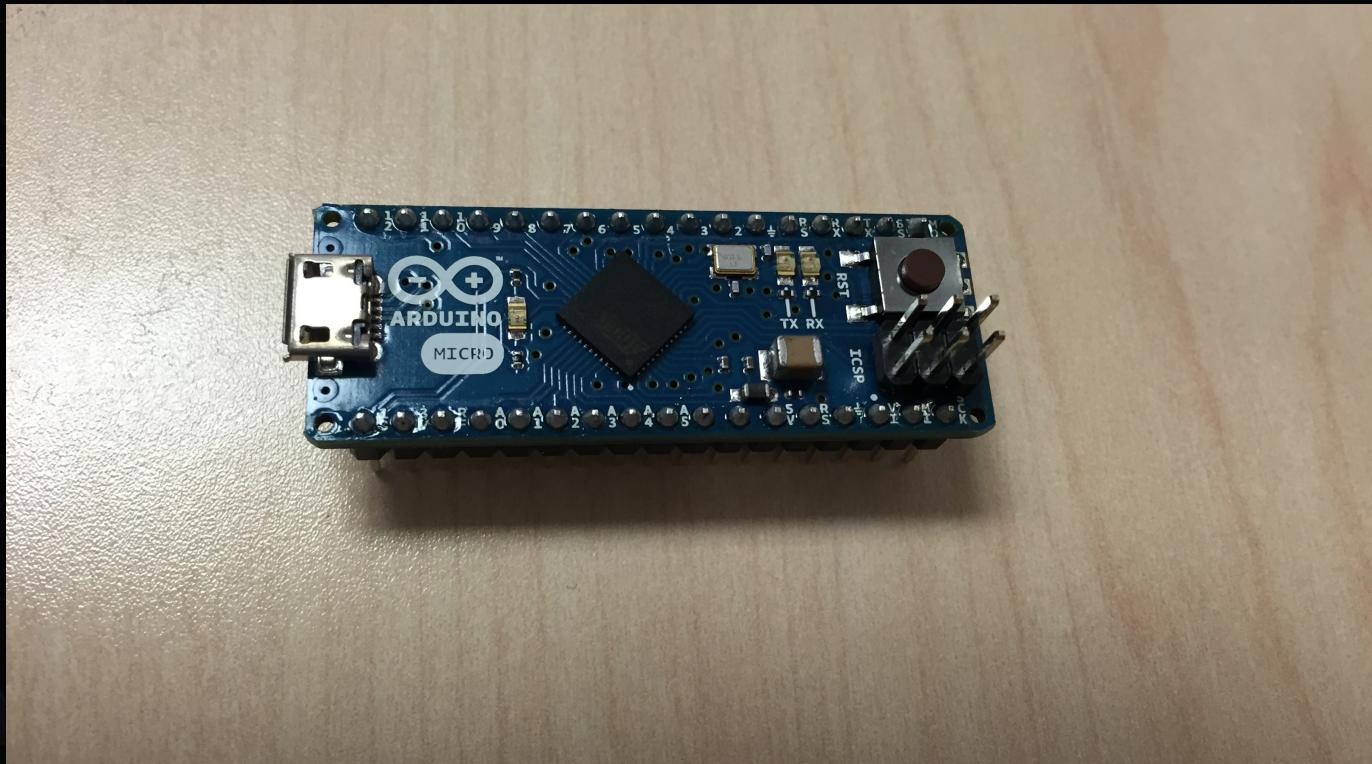
- ▶ Get experience with circuits
- ▶ Learn some basic C programming
- ▶ Get experience building things
- ▶ Inspiration to make

WHY BUILD OUR OWN GAME CONTROLLER?

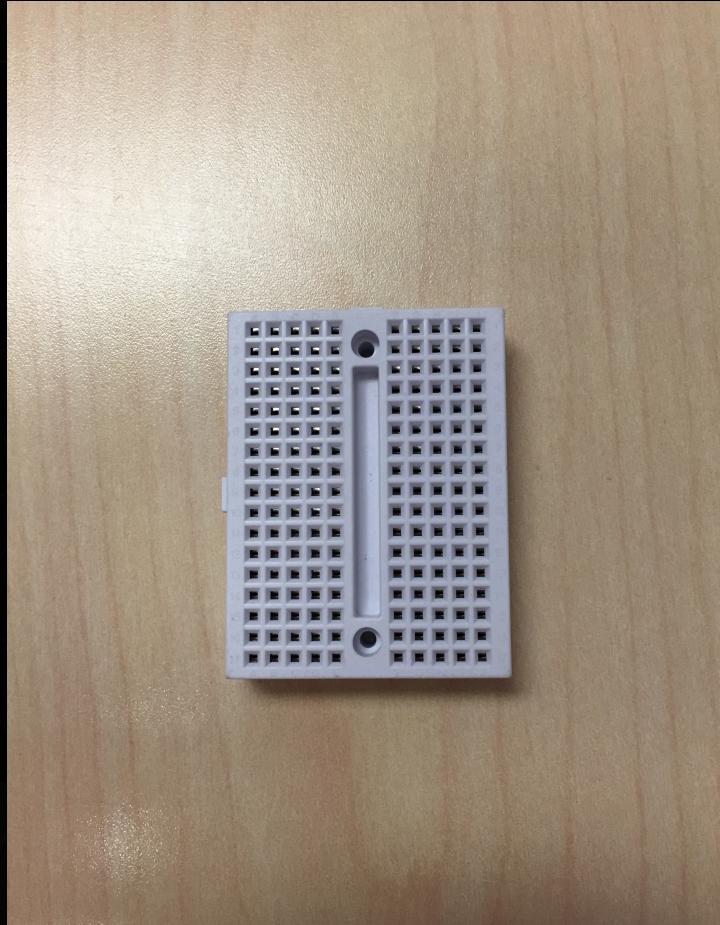
Selfishness

- ▶ Control over behavior
- ▶ Control over the feel
- ▶ Control over the look
- ▶ Control any game!
- ▶ Can add functionality

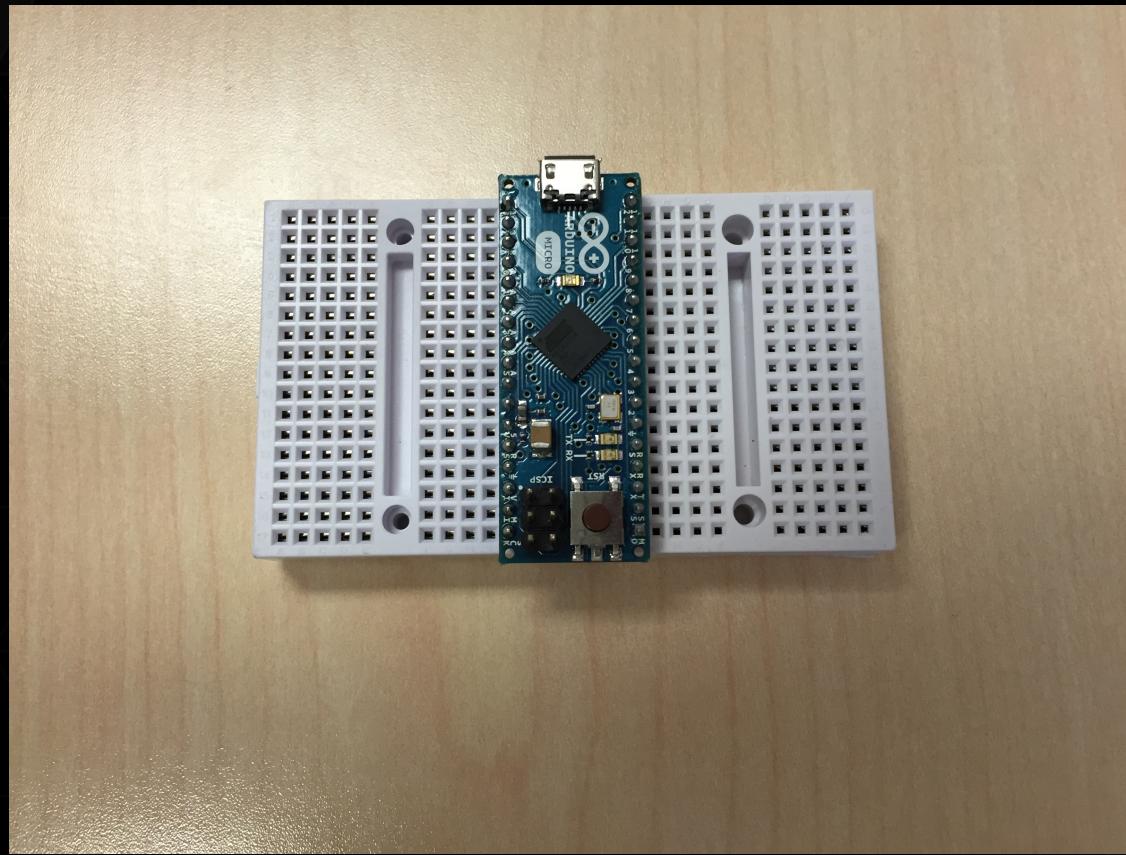
ARDUINO MICRO



BREADBOARDS



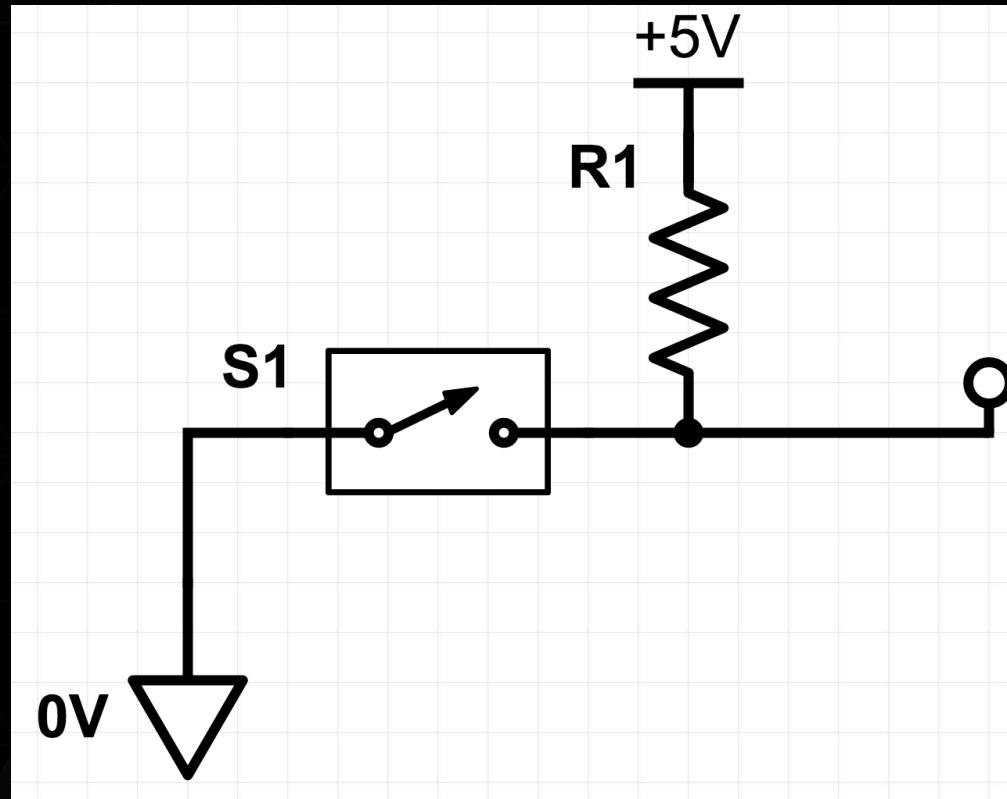
PUT ARDUINO IN BREADBOARDS



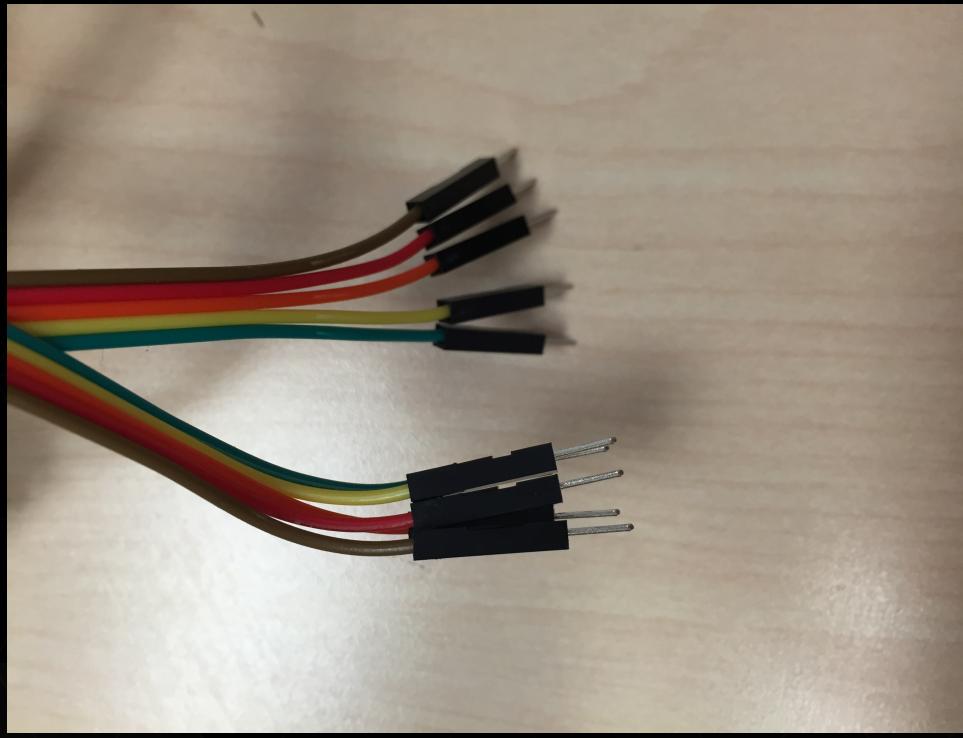
MAKE IT BLINK

```
// Josef Spjut
// Blink Example
void setup() {
    pinMode(13, OUTPUT);
}
void loop() {
    digitalWrite(13, HIGH);
    delay(500);
    digitalWrite(13, LOW);
    delay(500);
}
```

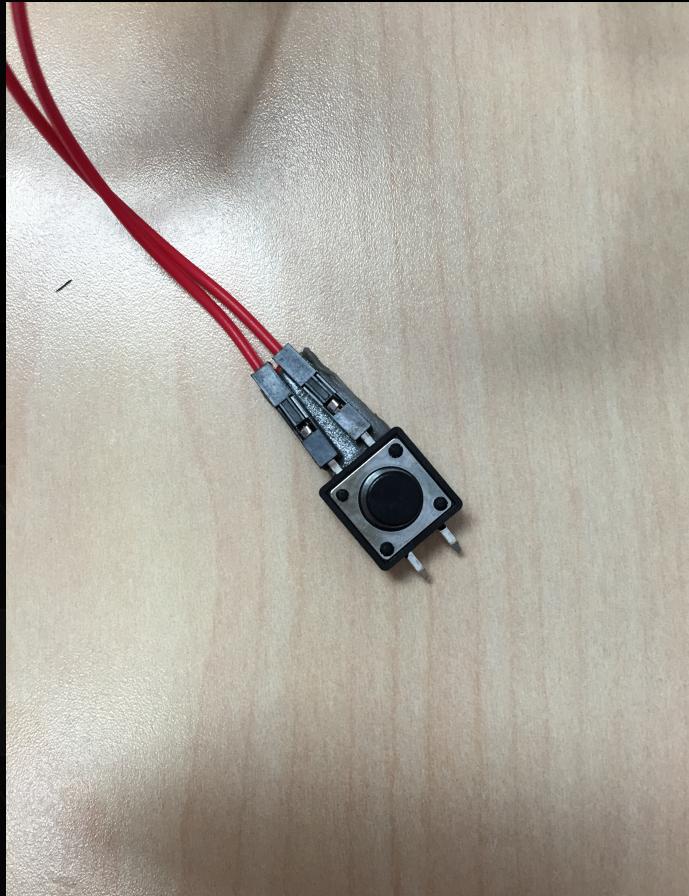
SWITCH



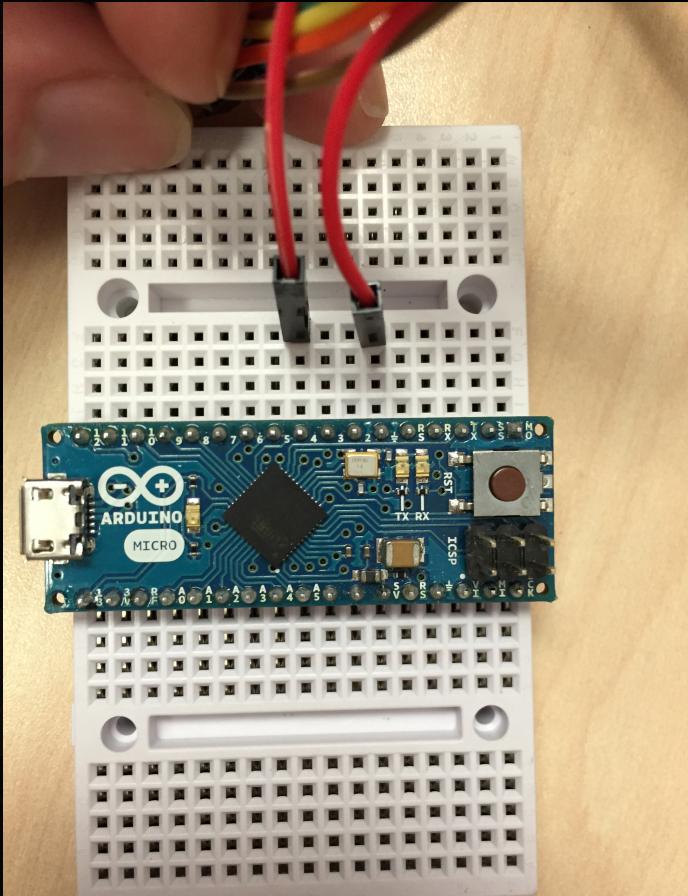
WIRES



BUTTON



WIRE BUTTON TO ARDUINO



- ▶ One side to Ground
- ▶ One side to Pin number

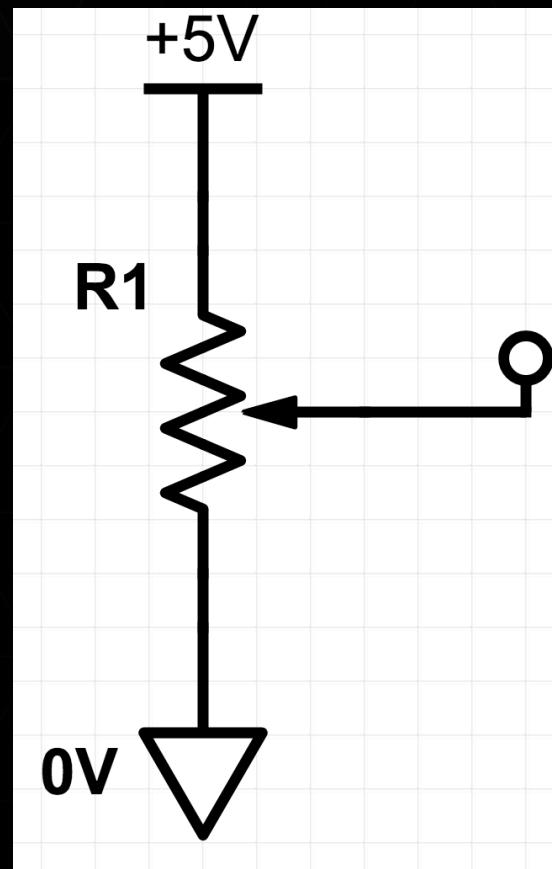
READ A BUTTON

```
// Josef Spjut
// Button Example
void setup() {
    pinMode(4, INPUT_PULLUP);      // set pullup resistor for button
    pinMode(13, OUTPUT);          // set LED out
}
void loop() {
    int val = digitalRead(4);      // Read button
    digitalWrite(13, val);         // Turn LED to match button
    delay(5);                    // wait 5 ms
}
```

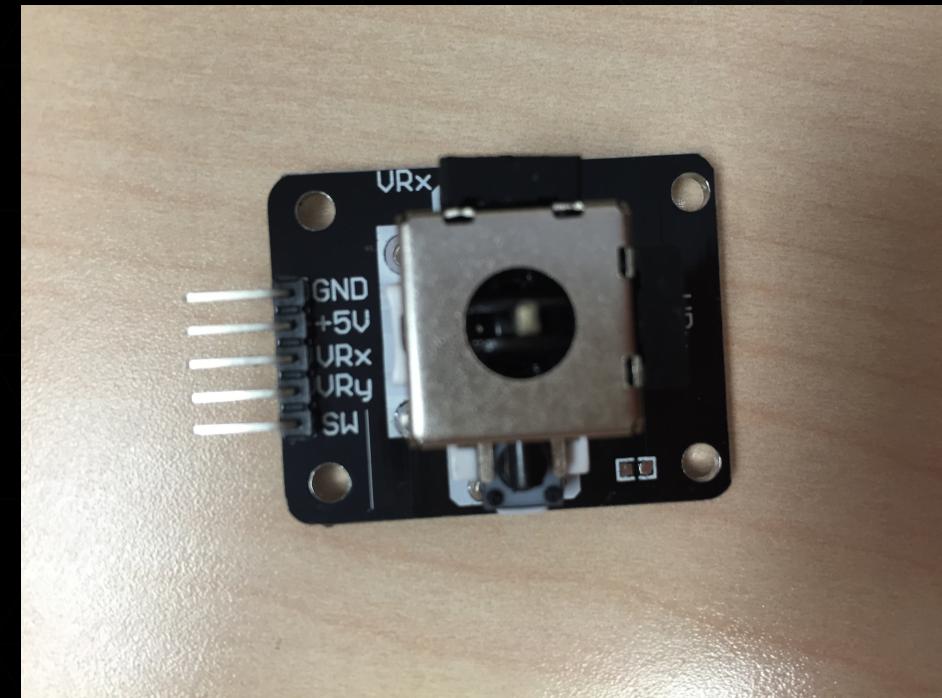
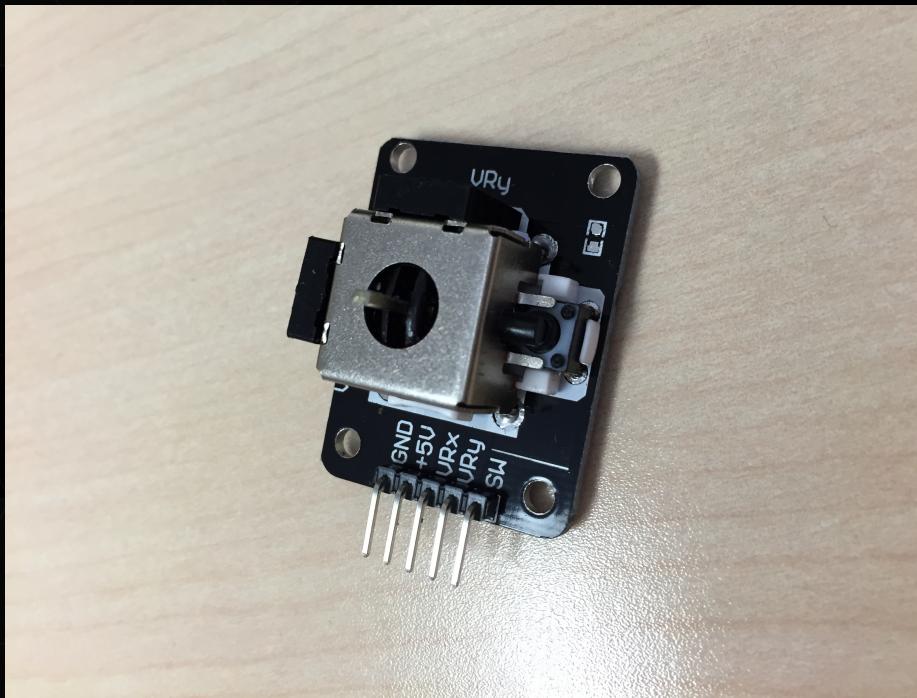
GENERIC POTENTIOMETER

- ▶ Circuit diagram
- ▶ Changes voltage at the middle pin linearly as you turn it

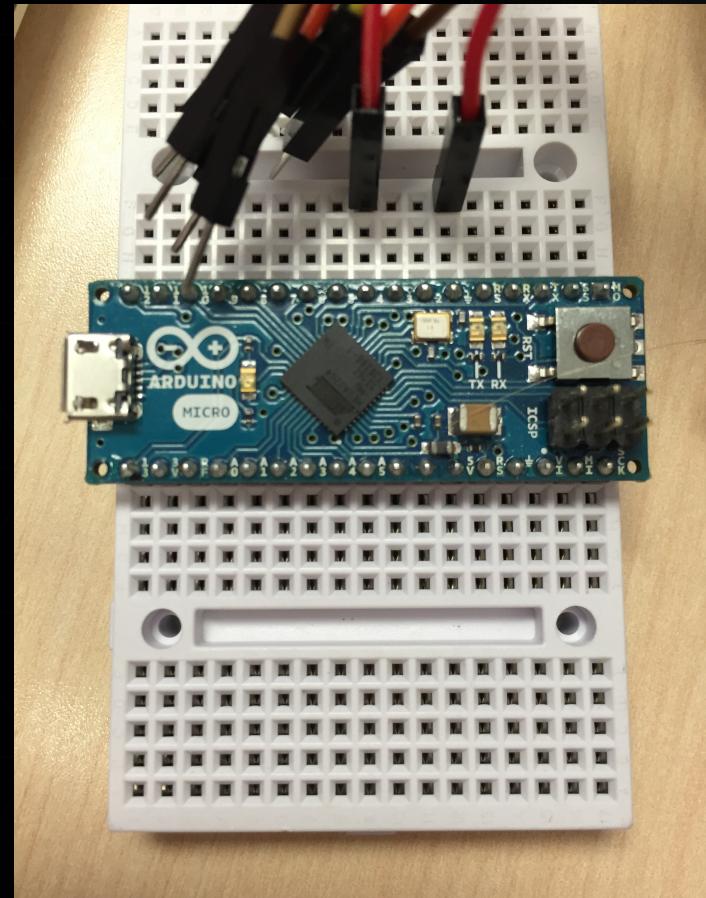
JOYSTICK POTENTIOMETER



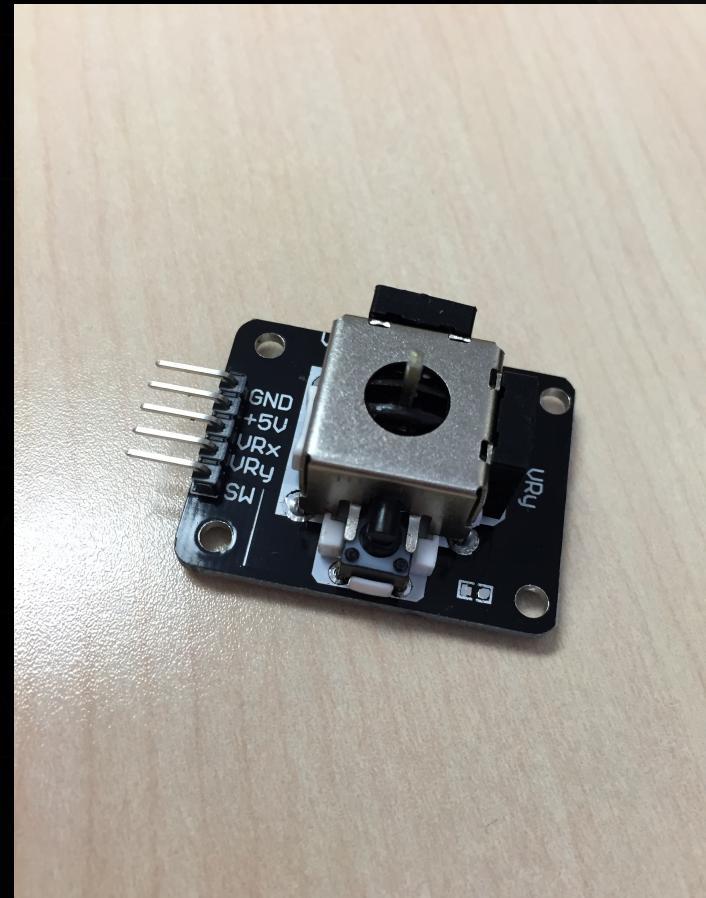
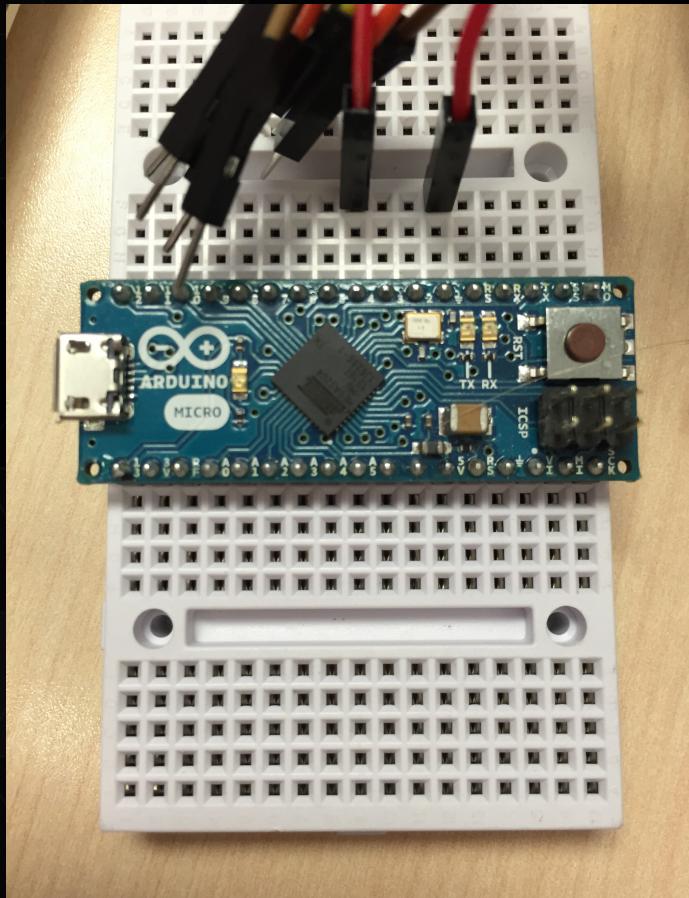
JOYSTICK POTENTIOMETER



WIRE JOYSTICK



CONNECT TO ARDUINO



JOYSTICK POTENTIOMETER

```
// Global variables  
const int xAxis = A2;  
const int yAxis = A4;  
  
// parameters for readAxis  
int responseDelay = 5;  
int range = 256;  
int threshold = range/4;  
int center = range/2;
```

JOYSTICK POTENTIOMETER

```
void setup(){  
    pinMode(13, OUTPUT);  
}  
  
void loop(){  
    int xReading = readAxis(xAxis, 0);  
    analogWrite(13, xReading);  
    delay(responseDelay);  
}
```

JOYSTICK POTENTIOMETER

```
int readAxis(int thisAxis, int isY) {  
    int reading = analogRead(thisAxis);          // Read analog input  
    reading = map(reading, 0, 1023, 0, range);    // map the reading from the analog input range  
    if (isY) reading = range-reading;            // invert y axis  
    // if the output reading is outside the rest position threshold  
    int distance = reading - center;  
    if (abs(distance) < threshold) {  
        distance = 0;  
    }  
    return distance;  
}
```

MOUSE LIBRARY

- ▶ Change range to 18 (sensitivity)
- ▶ `Mouse.begin()` in `init()`
- ▶ `Mouse.move(x, y, 0)`
- ▶ `Mouse.press()`
- ▶ `Mouse.release()`
- ▶ `Mouse.click()` - press and release at once

MOUSE INPUT

```
void setup() {  
    pinMode(13, OUTPUT);  
    Mouse.begin();  
}  
  
void loop() {  
    int xReading = readAxis(xAxis, 0);  
    int yReading = readAxis(yAxis, 0);  
    analogWrite(13, xReading);  
    Mouse.move(xReading, yReading, 0);  
    delay(responseDelay);  
}
```

KEYBOARD

- ▶ Keyboard.begin() // in init()
- ▶ Keyboard.press('w')
- ▶ Keyboard.release('w')
- ▶ Keyboard.print('w') - press and release at once

FLOOR PLAN

- ▶ Look at where you want to place things
- ▶ Trace locations with a pencil
- ▶ Remember your Arduino and breadboards!

MOUNT COMPONENTS



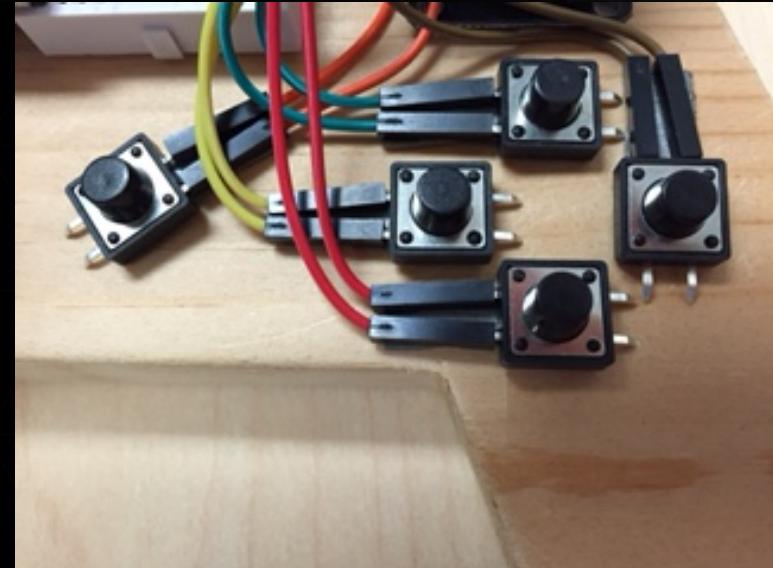
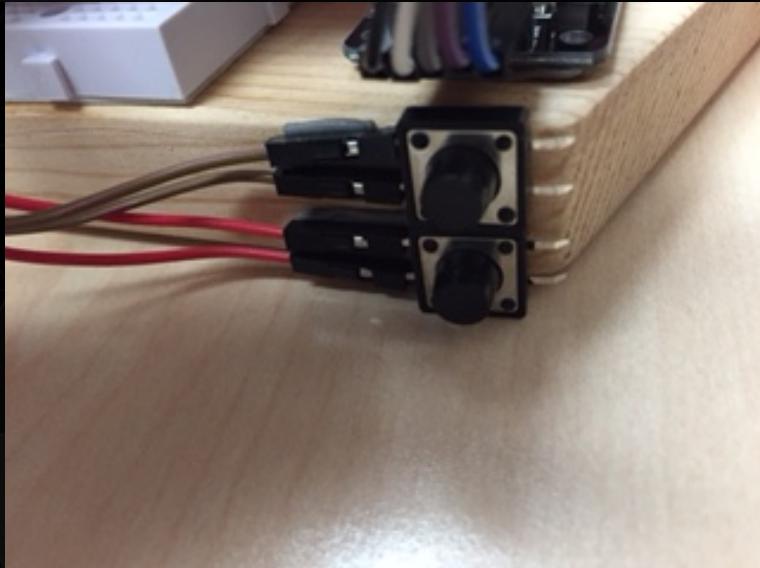
- ▶ Mount breadboards
- ▶ Mount Joysticks
- ▶ Stick on buttons

WIRING

- ▶ Connect things
- ▶ Soldering would be normal
- ▶ We're using 2-sided foam tape

MOUNT BUTTONS

Should, d-pad, Face buttons (your choice)



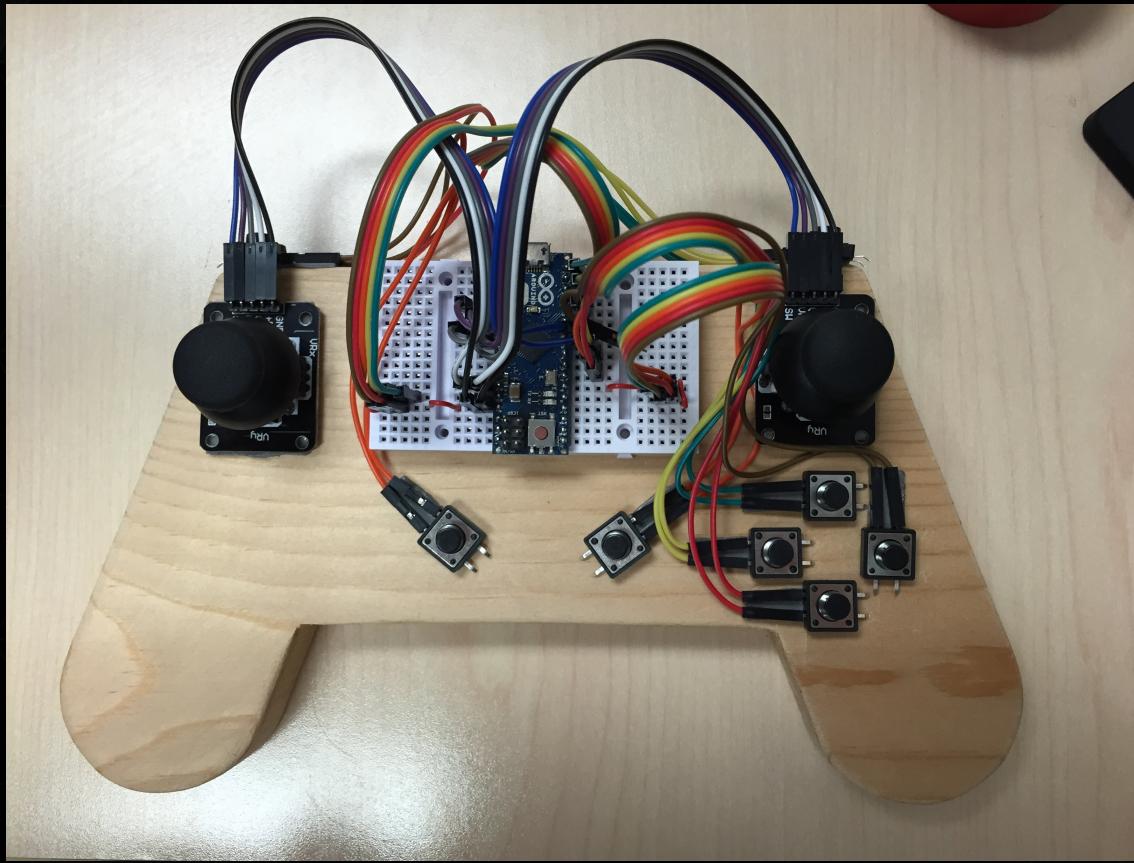
BRINGING IT ALL TOGETHER

- ▶ Program and you're good to go!
- ▶ Start a game and test

RECOMMENDED CONTROLLER PROGRAM

- ▶ Available at <http://josef.spjut.me/class/firmware.c>

FINISHED CONTROLLER



PLAYTESTING!

- ▶ Try out the mouse and keyboard input
- ▶ OpenArena
- ▶ Should work with Counterstrike, TF2, Battlefield, CoD and more!

MORE OPTIONS

- ▶ 3D print the controller base
- ▶ Hack an existing controller
- ▶ Custom Printed Circuit Board (PCB)
- ▶ Advanced firmware
- ▶ Wireless:
 - ▶ Bluetooth module (BluSmRF HID)
 - ▶ Battery pack

THANK YOU!

- ▶ Josef Spjut
- ▶ josef.spjut@gmail.com
- ▶ <http://josef.spjut.me>
- ▶ Other examples: <http://pages.hmc.edu/jspjut/class/s2015/e190u/>
- ▶ Feel free to contact me with any questions!

