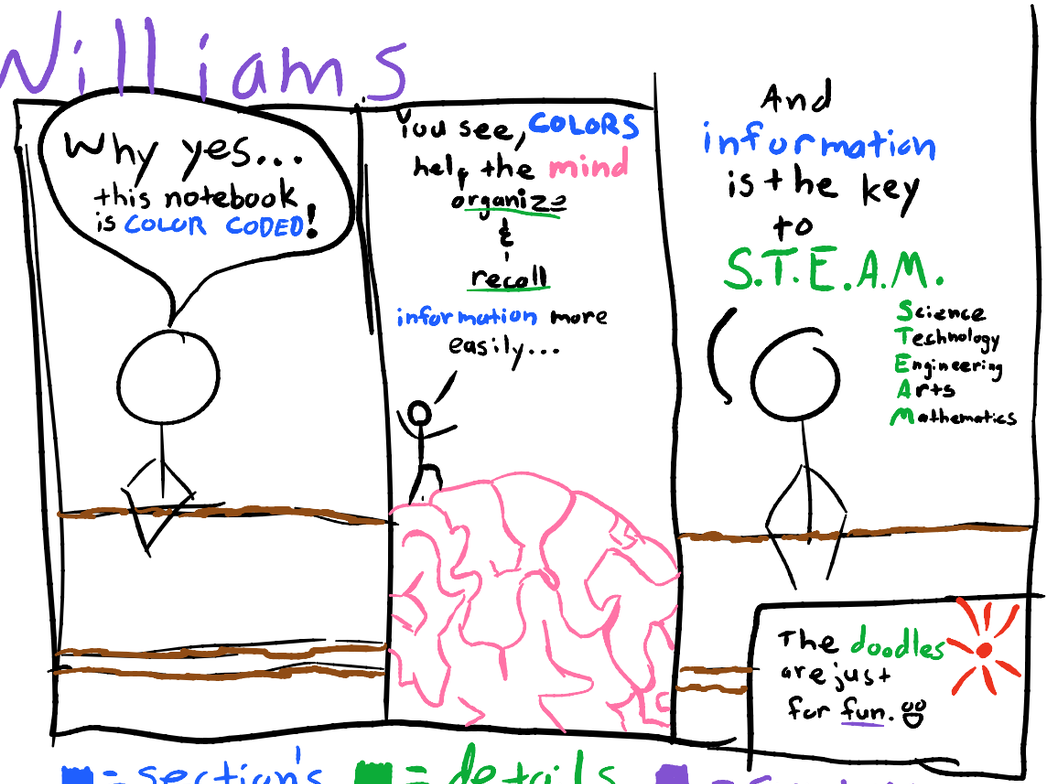
Ras Pi -SERIAL-> Arduino by Shane Williams

Monday, November 7, 2016 7:45 PM



Arduino

Arduino Uno
Laptop

- the SERIAL LIBRARY can be used to communicate with other SERIAL devices.
- I'll be using the Arduino's USB port to communicate b/w it and a host computer

In this notebook...

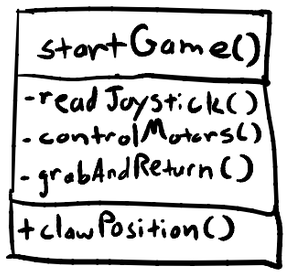
- Serial

...Some of the functions I'm considering from the SERIAL LIBRARY...

- Serial.readBytesUntil()
- Serial.setTimeout()
- Serial.readString()
- Serial.begin()
- Serial.end()
- Serial.println()
- Serial.readStringUntil()
- Serial.print()

Serial
digitalWrite
Strings & arrays
interrupts

- input + output

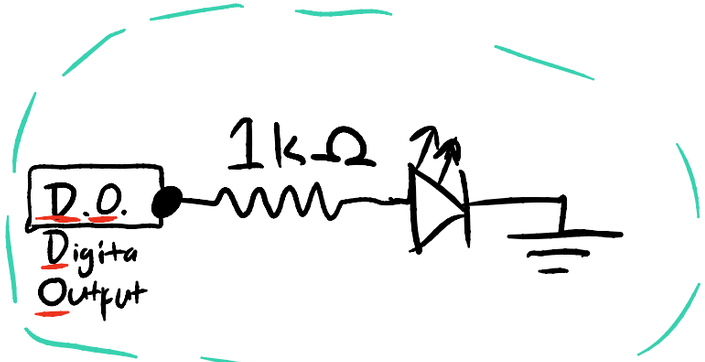


So to trouble shoot my process, I'll be using the Arduino's serial monitor.

the Arduino's serial monitor.

BUT, only one process can access this at a time.

- Plan (at first thought) is to use digital output pins on the Arduino to light an LED bulb when a button is pressed



Basic circuit idea...

  - We are reading 1's & 0's, but digitalWrite(pin, value) requires HIGH & LOW
    - ~~will need to convert string of 1's & 0's to array of HIGH & LOW~~

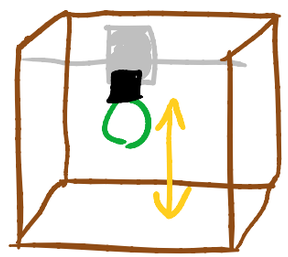
NOT SO. SEE BELOW (13Now)

  - Will also be reading floating point values.

Servos (3)
- controlled by PWM

How can we tell how far down the servo has gone?

Stepper Motors (2)
- controlled by:
  - Direction
  - Enable
  - Square wave



floating point values. Matthew had the idea of multiplying these by 100 (or 1000) so we can send as an integer and divide by 100 again in the Arduino to get back to float.

Square wave 

IDEA:

When reading the button value string, set an interrupt to trigger the section of code that drops the claw, grabs the "stuff," and return back to home position

---

13 Nov 2016

ALRIGHT.

- So I've got the Arduino writing to digital Outputs pins based on

pinWriter

```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);               // open serial port. Baud rate is 9600
  for (int i=2;i<=9;i++)
  {
    pinMode(i, OUTPUT);
  }
}

```

- So I've got the ... to digitalOutput pins based on a constant character variable I've hard-coded. Eventually this will be created from a Serial read event instead

  char bitValues[8]={1,0,1,1,0,1,0,0};

  (lines 14 & 13)

- Commented-out lines 23-25 will write a string to a character array
  - 26 & 27 are used to verify we are sending the right stuff to digitalWrite()

- NEXT STEP is to put whatever is read from a serial read command into the format above.
  - would be best to use joystick's axis data to drive PWM & other pins. This will give real-time

```
6     pinMode(i, OUTPUT);
7   }
8 }
9
10 void loop() {
11   // put your main code here, to run repeatedly:
12   int it = 0;
13 //String bitValues[8] = {"HIGH", "LOW", "HIGH", "HIGH", "LOW", "HIGH", "LOW", "LOW"};
14 // char bitValues[8] = {1,0,1,1,0,1,0,0};
15 char bitValues[8] = {0,1,0,0,1,0,1,1};
16 while (it<8)
17 {
18   Serial.println(bitValues[it]);
19   // Serial.println(it);
20   //Serial.println(it);
21   delay(500);
22   int pin = 2 + it;
23   //int strLen = bitValues[it].length()+1;
24   //char charArray[strLen];
25   //bitValues[it].toCharArray(charArray, strLen);
26   //Serial.print(pin);
27   //Serial.println(charArray);
28   digitalWrite(pin, bitValues[it]);
29   it++;
30
31   }
32 it=0;
33 delay(2500);
34 while (it<8)
35 {
36   int pin=2+it;
37   digitalWrite(pin, LOW);
38   it++;
39 }
40 Serial.println("Waiting...");
41 delay(2000);
42 }
```

drive PWM & other pins. This will give real-time feedback w/o requiring use of the serial monitor.

---

15 Nov 2016

TJ had an IDEA! Use random numbers to vary servo strength so they don't always win!!

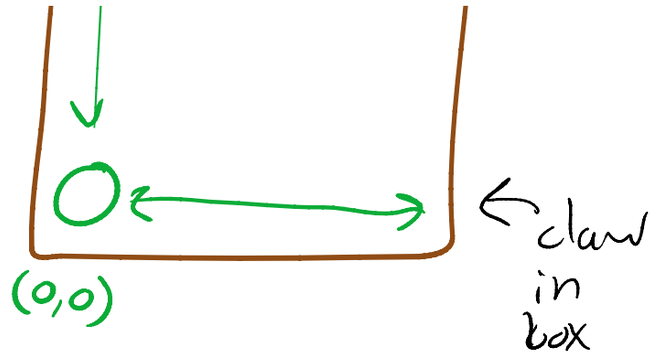
Motor Controllers

- Step the motor every 2 pulses
  - send a square wave
  - higher frequency = faster stepping
  - 'joystick' sets frequency
- Determining Claw Position

Enable
Direction
Waveform
} Input pins for M.C.

determining [illegible]

- Can find # of pulses, use that to determine the claw's position
  - also use this to bring it "home"



---

18 Nov 2016

What else needs to be done on software side...

- Set arduino to parse hex values to ints for motor control
- Python code to use event handlers (interrupts) for button presses and axes movements?
- Send axes and buttons in one string or only send

- Converting to hex: how can we maintain that "0-1" range for axes across conversions
- planning on using interrupts to
  - quit program
  - start "Drop & Grab"
- Have python script add newline

send axes and buttons in one string or only send button data when pressed/released (using events)?

(1) Use fixed length for axes, buttons,

—or—

(2) Send seperately and prepend "a" for axes, "b" for buttons?

add newline character to string written to Arduino

Eric & Jacob

MOVE MOTORS

- reading axes' values
- record # of steps motors have taken in each axis
- sending control signals (direction (0,1), frequency [(0:1) · maxfreq])

Shane

DROP & GRAB

- use interrupt from button press
- drop claw(servo, PWM signal)
- close claw(" , ")
- raise claw(" , ")

(direction (0,1), frequency [(0:1) · maxfreq])

- Need to use PWM pins

- raise claw (" , ")
- return home (uses current position variable)
- Use random # so they don't always win

GAME OVER

- Display something (win/Lose) after drop & grab

Anyone (this is non-functional & of secondary concern A.T.M.)