

```
/*arduino pro micro 24 input Button box with 2 encoders and 5 fuction led
porgramed to be compatible with all devices that have keyboard
compatibility
this sketch was created/alterd and designed
by Carlos Reyes
with help from the arduino community and libraries
on august 2016
for the purpose of DIY project on youtube channel
metadata is still in progress so search
2727clos, carlos reyes, clos2727, clos reyes, or game on
email clos2727@yahoo.com */
```

```
//version 1.00    encoders confirmed and working
//version 1.50,   added buttons to sequential shifter
//version 2.00    removed shifter added keyboard press
//version 2.50    added 5 led lights
//version 3.00    debugging buttons and led function
//version 3.50    button presses are synced with led lights (final)
```

```
// button box version    V3.50
//
```

```
/* note
 * if using leonardo be sure to change pin numbers as pin numbers on pro
micro differ.
 */
```

```
#include <Keypad.h>
#include <Keyboard.h>
```

```
#define ENABLE_PULLUPS
#define NUMROTARIES 2
#define NUMBUTTONS 24
#define NUMROWS 5
#define NUMCOLS 5
```

```
//define the symbols on the buttons of the keypads
char buttons[NUMROWS][NUMCOLS] = {
    {'a','b','c','d'},      // 4      // col pins
    {'e','f','g','h'},      // 5
    {'i','j','k','l'},      // 6
    {'m','n','o','p'},      // 7
    {'q','r','s','t'}       // 8
    // row pins
    // 9  14  15  16
};
```

```
struct rotariesdef {
    byte pin1;
    byte pin2;
    int ccwchar;
    int cwchar;
```

```

    volatile unsigned char state;
};
//{encoder pins,characters press
rotariesdef rotaries[NUMROTARIES] {
    {0,1,'u','v',},          // encoder connected to pin 0 and 1 middle pin is
ground
    {2,3,'w','x',},          // encoder connected to pin 2 and 3 middle pin is
ground
};

#define ledpinA10 A10
#define ledpinA0 A0
#define ledpinA1 A1
#define ledpinA2 A2
#define ledpinA3 A3

#define DIR_CCW 0x10
#define DIR_CW 0x20

#define R_START 0x0
#define R_CW_FINAL 0x1
#define R_CW_BEGIN 0x2
#define R_CW_NEXT 0x3
#define R_CCW_BEGIN 0x4
#define R_CCW_FINAL 0x5
#define R_CCW_NEXT 0x6

boolean ledA0_state;
boolean ledA1_state;
boolean ledA2_state;
boolean ledA3_state;
boolean ledA7_state;
boolean ledA10_state;

unsigned long loopCount;
unsigned long startTime;
String msg;

const unsigned char ttable[7][4] = {
    // R_START
    {R_START,    R_CW_BEGIN,    R_CCW_BEGIN, R_START},
    // R_CW_FINAL
    {R_CW_NEXT,  R_START,        R_CW_FINAL,  R_START | DIR_CW},
    // R_CW_BEGIN
    {R_CW_NEXT,  R_CW_BEGIN,    R_START,      R_START},
    // R_CW_NEXT
    {R_CW_NEXT,  R_CW_BEGIN,    R_CW_FINAL,   R_START},
    // R_CCW_BEGIN
    {R_CCW_NEXT, R_START,        R_CCW_BEGIN, R_START},
    // R_CCW_FINAL
    {R_CCW_NEXT, R_CCW_FINAL,    R_START,      R_START | DIR_CCW},

```

```

    // R_CCW_NEXT
    {R_CCW_NEXT, R_CCW_FINAL, R_CCW_BEGIN, R_START},
};

byte rowPins[NUMROWS] = {4,5,6,7,8,}; //connect to the row pinouts of the
keypad
byte colPins[NUMCOLS] = {9,14,15,16,}; //connect to the column pinouts of
the keypad

//initialize an instance of class NewKeypad
Keypad buttbx = Keypad( makeKeymap(buttons), rowPins, colPins, NUMROWS,
NUMCOLS);

void setup() {
// initialize all analog pins as output
pinMode(ledpinA1,OUTPUT);
pinMode(ledpinA0,OUTPUT);
pinMode(ledpinA2,OUTPUT);
pinMode(ledpinA3,OUTPUT);
pinMode(ledpinA10,OUTPUT);

analogWrite(ledpinA0,128);
analogWrite(ledpinA1,128);
analogWrite(ledpinA2,128);
analogWrite(ledpinA3,128);
analogWrite(ledpinA10,128);

digitalWrite(ledpinA3,HIGH);
digitalWrite(ledpinA0,HIGH);
digitalWrite(ledpinA2,HIGH);
digitalWrite(ledpinA1,HIGH);
digitalWrite(ledpinA10,HIGH);

delay(200);
digitalWrite(ledpinA3,LOW);
digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA10,LOW);

delay(200);
digitalWrite(ledpinA3,HIGH);
digitalWrite(ledpinA0,HIGH);
digitalWrite(ledpinA1,HIGH);
digitalWrite(ledpinA2,HIGH);
digitalWrite(ledpinA10,HIGH);

delay(200);
digitalWrite(ledpinA10,LOW);

```

```
digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA3,LOW);

delay(200);

// led final setup all led stay on
digitalWrite(ledpinA3,HIGH);
delay(300);
digitalWrite(ledpinA10,HIGH);
delay(300);
digitalWrite(ledpinA2,HIGH);
  delay(300);
digitalWrite(ledpinA1,HIGH);
delay(300);
digitalWrite(ledpinA0,HIGH);
delay(300);
```

```
digitalWrite(ledpinA10,LOW);
digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA3,LOW);
```

```
delay(200);
```

```
digitalWrite(ledpinA1,HIGH);
digitalWrite(ledpinA2,HIGH);
digitalWrite(ledpinA0,HIGH);
digitalWrite(ledpinA10,HIGH);
digitalWrite(ledpinA3,HIGH);
```

```
delay(200);
digitalWrite(ledpinA10,LOW);
digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA3,LOW);
```

```
delay(200);
digitalWrite(ledpinA3,HIGH);
digitalWrite(ledpinA0,HIGH);
digitalWrite(ledpinA1,HIGH);
digitalWrite(ledpinA2,HIGH);
digitalWrite(ledpinA10,HIGH);
```

```
delay(200);
digitalWrite(ledpinA10,LOW);
```

```

digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA3,LOW);

delay(200);
// led final setup all led stay on
digitalWrite(ledpinA3,HIGH);
delay(200);
digitalWrite(ledpinA10,HIGH);
delay(200);
digitalWrite(ledpinA2,HIGH);
delay(200);
digitalWrite(ledpinA1,HIGH);
delay(200);
digitalWrite(ledpinA0,HIGH);
delay(200);

digitalWrite(ledpinA10,LOW);
digitalWrite(ledpinA0,LOW);
digitalWrite(ledpinA1,LOW);
digitalWrite(ledpinA2,LOW);
digitalWrite(ledpinA3,LOW);

digitalWrite(ledpinA10,HIGH);
digitalWrite(ledpinA0,HIGH);
digitalWrite(ledpinA1,HIGH);
digitalWrite(ledpinA2,HIGH);
digitalWrite(ledpinA3,HIGH);


Keyboard.begin();
rotary_init();
}
void loop() {

    CheckAllEncoders();

    CheckAllButtons();
}
void CheckAllButtons(void) {
    char key = buttbx.getKey();

    if (key == 'a') {
        Keyboard.write(key);
        digitalWrite(ledpinA10,LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA10,HIGH);
    }
}

```

```

}

if (key == 'b') {
    Keyboard.write(key);
    digitalWrite(ledpinA10,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA10,HIGH);
}
    if (key == 'c') {
        Keyboard.write(key);
        digitalWrite(ledpinA10,LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA10,HIGH);
    }
if (key == 'd') {
    Keyboard.write(key);
    digitalWrite(ledpinA10,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA10,HIGH);
}
if (key == 'e') {
    Keyboard.write(key);
    digitalWrite(ledpinA10,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA10,HIGH);
}
if (key == 'f') {
    Keyboard.write(key);
    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}

    if (key == 'g'){
Keyboard.write('g');
digitalWrite(ledpinA10,LOW);
delay(150);
Keyboard.release('g');
digitalWrite(ledpinA10,HIGH);
    }
if (key == 'h') {
    Keyboard.write(key);
    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}
if (key == 'i') {
    Keyboard.write(key);

```

```

    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}
if (key == 'j') {
    Keyboard.write(key);
    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}
if (key == 'k') {
    Keyboard.write(key);
    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}
if (key == 'l') {
    Keyboard.write(key);
    digitalWrite(ledpinA3,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA3,HIGH);
}
if (key == 'm') {
    Keyboard.write(key);
    digitalWrite(ledpinA2,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA2,HIGH);
}
if (key == 'n') {
    Keyboard.write(key);
    digitalWrite(ledpinA1,LOW);
    delay(150);
    Keyboard.release(key);
    digitalWrite(ledpinA1,HIGH);
}
if (key == 'o') {
    Keyboard.write(key);
    digitalWrite(ledpinA1,LOW);
    delay(150);
    digitalWrite(ledpinA1,HIGH);
    Keyboard.release(key);
}
if (key == 'p') {
    Keyboard.write(key);
    digitalWrite(ledpinA1,LOW);
    delay(150);
    digitalWrite(ledpinA1,HIGH);
    Keyboard.release(key);
}

```

```

    }

    if (key == 'q') {
        Keyboard.write(key);
        digitalWrite(ledpinA1, LOW);
        delay(150);
        digitalWrite(ledpinA1, HIGH);
        Keyboard.release(key);
    }
    if (key == 'r') {
        Keyboard.write(key);
        digitalWrite(ledpinA1, LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA1, HIGH);
    }
    if (key == 's') {
        Keyboard.write(key);
        digitalWrite(ledpinA1, LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA1, HIGH);
    }
    if (key == 't') {
        Keyboard.write(key);
        digitalWrite(ledpinA1, LOW);
        delay(150);
        digitalWrite(ledpinA1, HIGH);
        Keyboard.release(key);
    }
    /*if (key == 's') {
        Keyboard.write(key);
        digitalWrite(ledpinA10, LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA10, HIGH);
    }
    if (key == 't') {
        Keyboard.write(key);
        digitalWrite(ledpinA2, LOW);
        delay(150);
        Keyboard.release(key);
        digitalWrite(ledpinA2, HIGH);
    }*/

}

/* Call this once in setup(). */
void rotary_init() {
    for (int i=0; i<NUMROTARIES; i++) {
        pinMode(rotaries[i].pin1, INPUT);
        pinMode(rotaries[i].pin2, INPUT);
    }
}

```



```

        #ifdef ENABLE_PULLUPS
            digitalWrite(rotaries[i].pin1, HIGH);
            digitalWrite(rotaries[i].pin2, HIGH);
        #endif
    }
}

/* Read input pins and process for events. Call this either from a
 * loop or an interrupt (eg pin change or timer).
 *
 * Returns 0 on no event, otherwise 0x80 or 0x40 depending on the
direction.
 */
unsigned char rotary_process(int _i) {
    unsigned char pinstate = (digitalRead(rotaries[_i].pin2) << 1) |
digitalRead(rotaries[_i].pin1);
    rotaries[_i].state = ttable[rotaries[_i].state & 0xf][pinstate];
    return (rotaries[_i].state & 0x30);
}

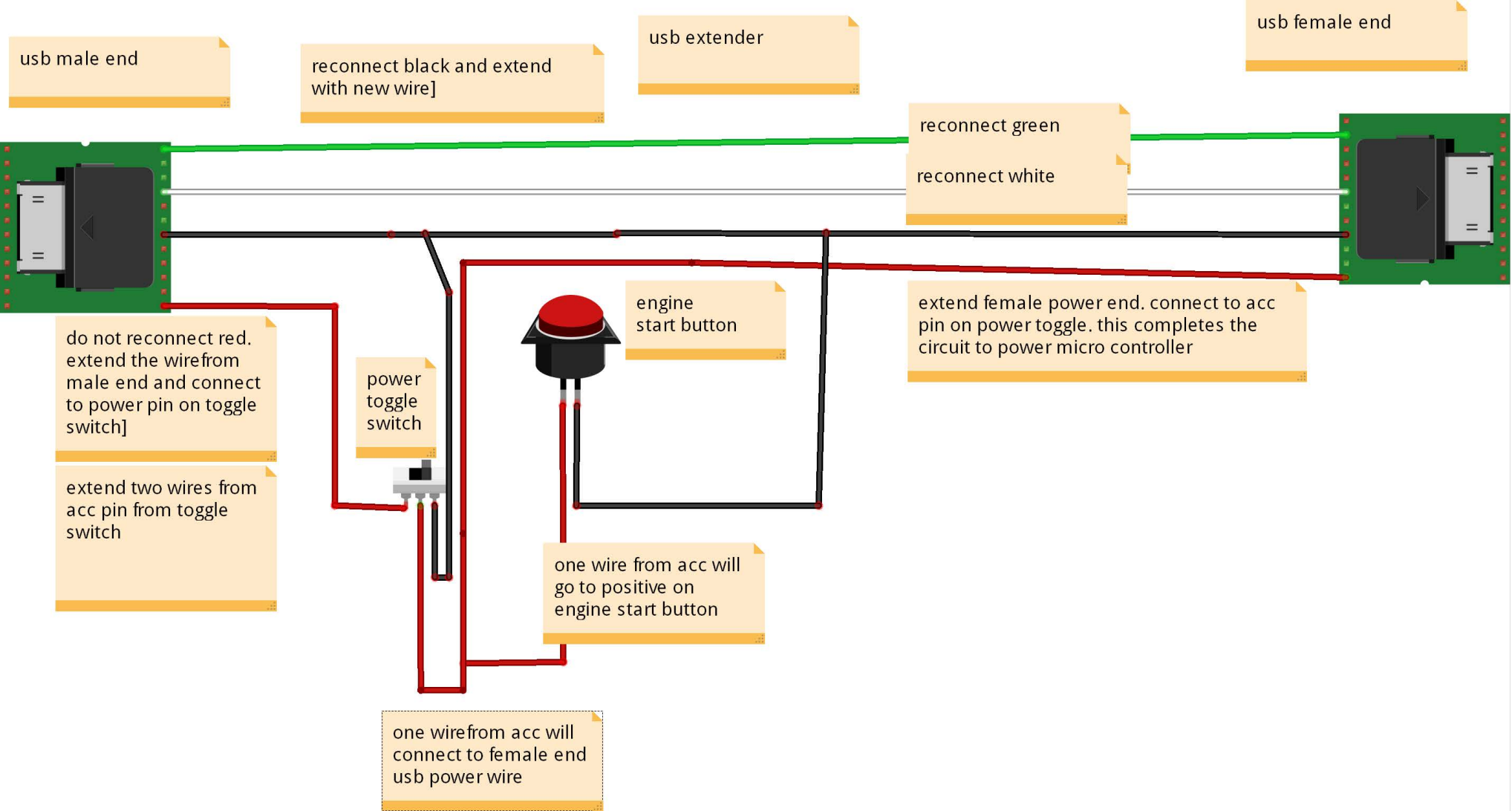
void CheckAllEncoders(void) {
    for (int i=0;i<NUMROTARIES;i++) {
        unsigned char result = rotary_process(i);
        if (result) {
            Keyboard.write(result == DIR_CCW ? rotaries[i].ccwchar :
rotaries[i].cwchar );

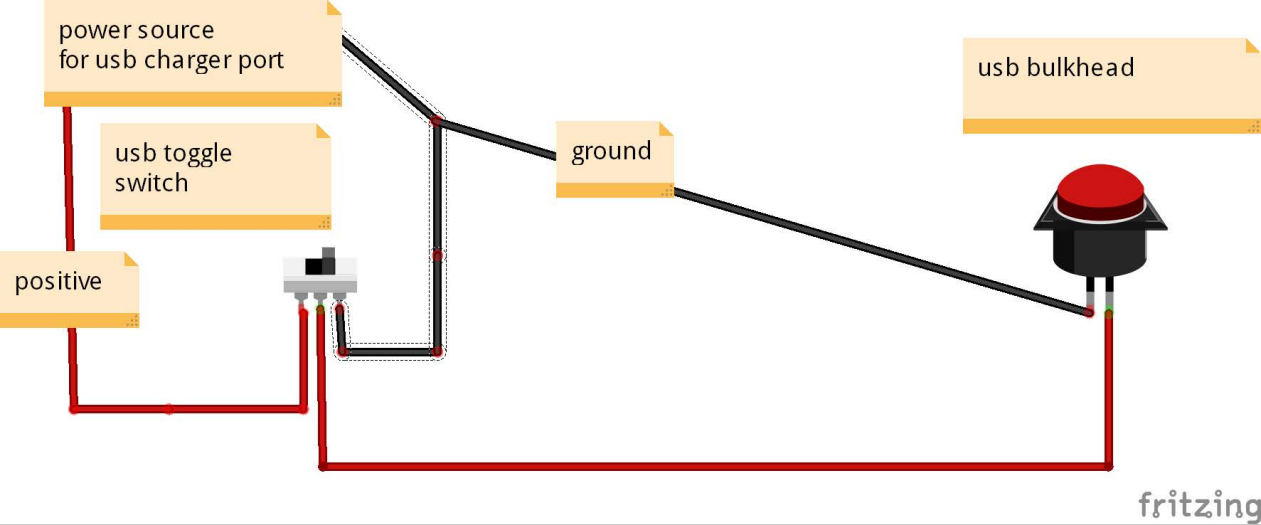
            //detects array characters assigned to a specific encoders and
blinks led
            // ccw = counter clockwise
            //cw =clock
            // (rotaries[i].ccwchar being used below in the if statement,
            //will call all characters in ccwchar scope.
            if(rotaries[i].ccwchar == 'u'){ // blink led from any
rotation
                digitalWrite(ledpinA0,LOW);
                delay(50);
                digitalWrite(ledpinA0,HIGH);
                delay(50);
            }
            if(rotaries[i].ccwchar == 'w'){
                digitalWrite(ledpinA0,LOW);
                delay(50);
                digitalWrite(ledpinA0,HIGH);
                delay(50);
            }

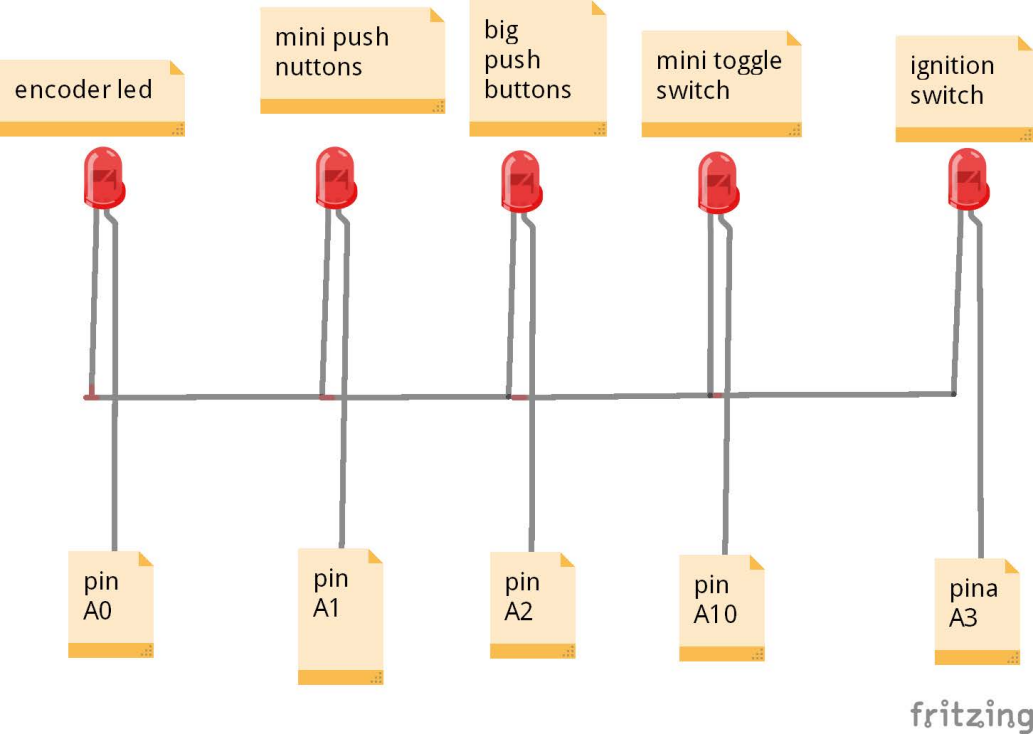
        }
    }
}

```

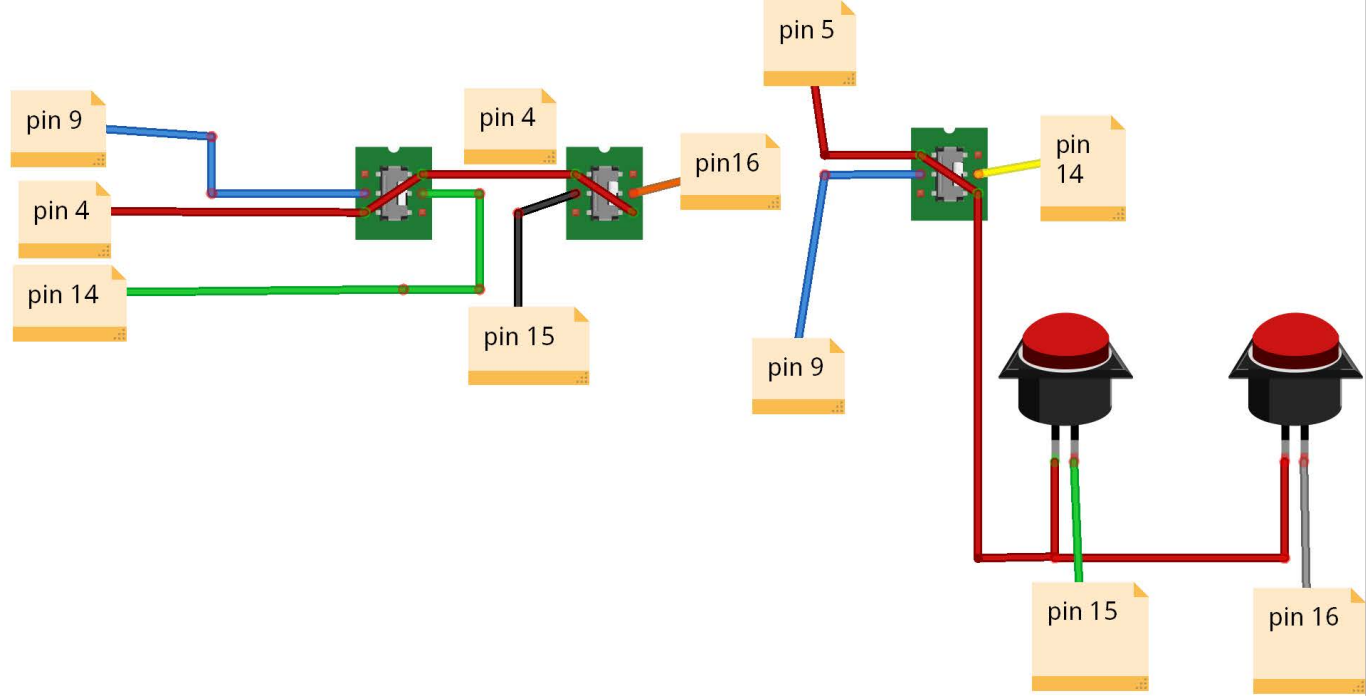
}

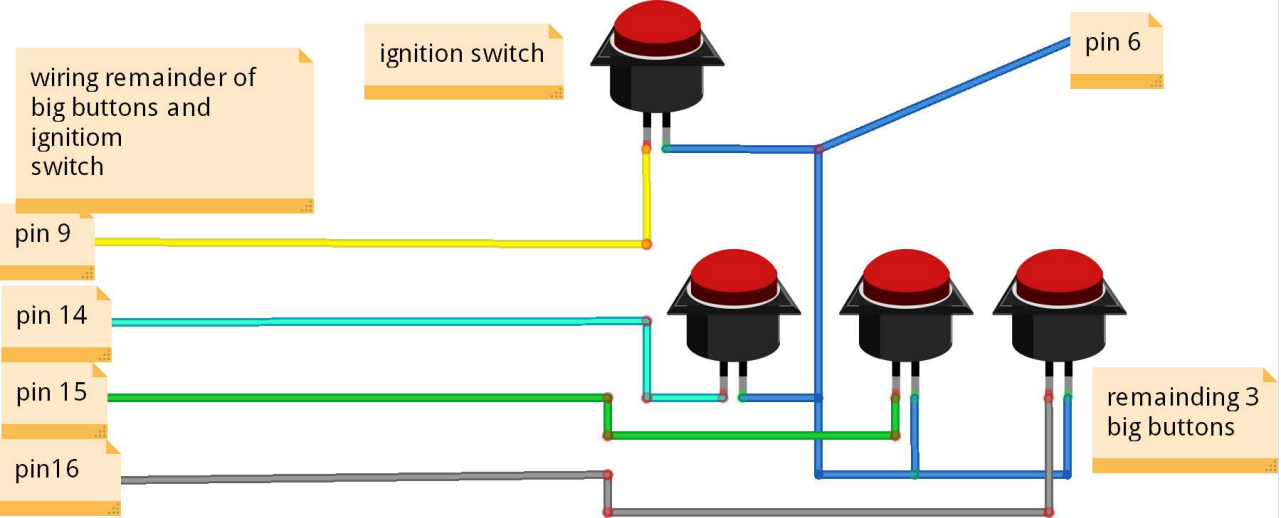






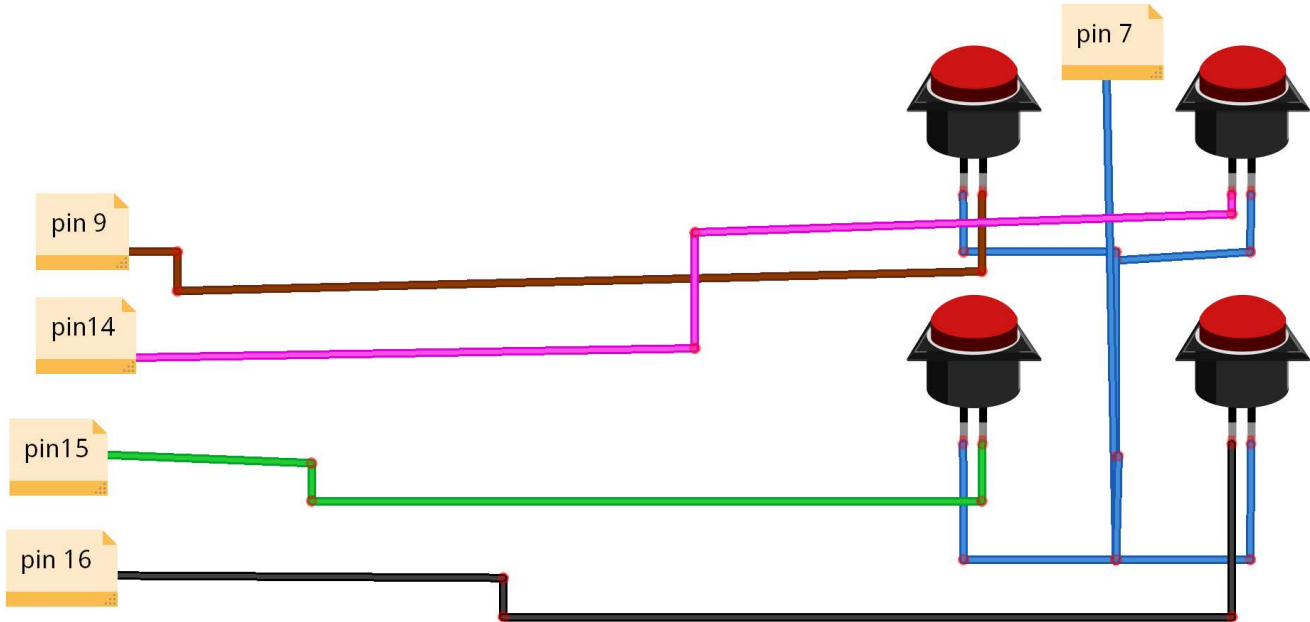
wiring dpdt switch
from left to right. and 2
big buttons

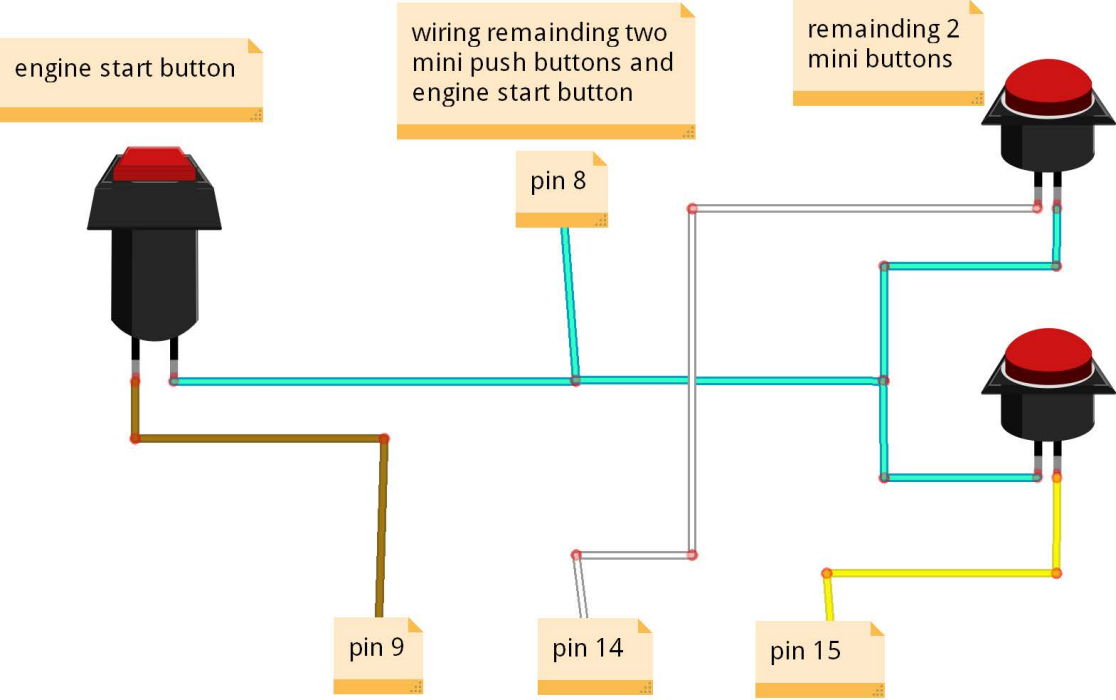




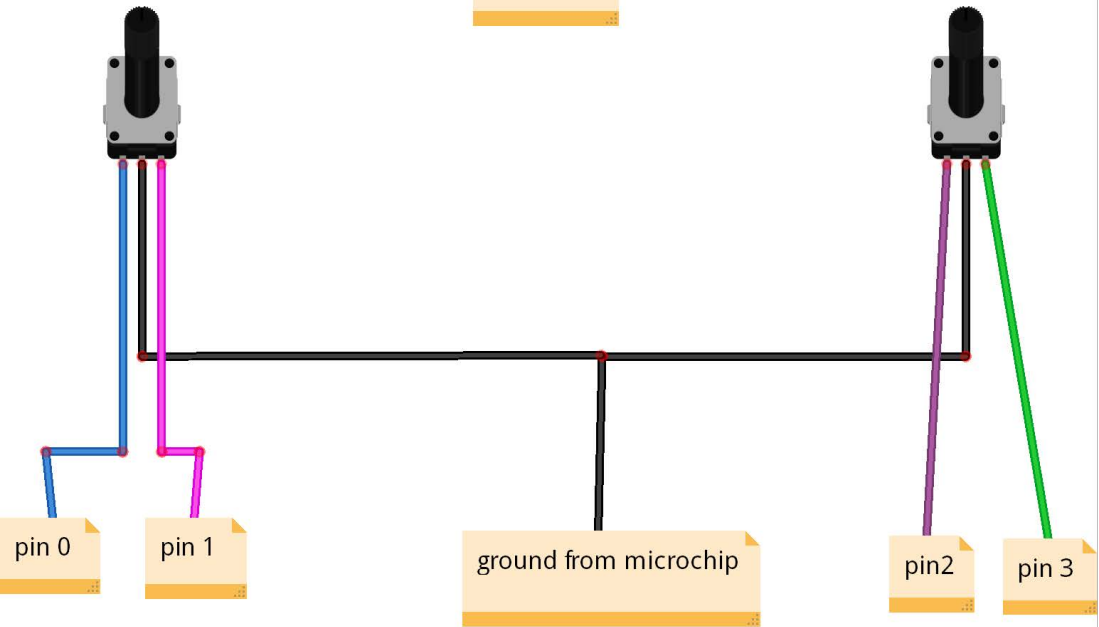
pin 7 inputs are all
used up.

wiring 4 mini
pushbutton from right
to left.





encoders



there is one input left over. if you want to use it, wire it to one of the encoder buttons opposite pins

