

Make: DIY Arcade Controller

CODE:

```
//Tyler Capps
//MAKE:
//April 2023
// Controlling an Arduino with an arcade style joystick and buttons
//selecting pins for each directional input
int dirDown =4;
int dirUp =5;
int dirRight =7;
int dirLeft =6;
//selecting pins for the output of our buttons
int buttonRed =8;
int buttonBlue =9;
int buttonGreen =10;
int buttonYellow =11;
int buttonValue;
void setup() {
// put your setup code here, to run once:
pinMode( dirDown , INPUT_PULLUP);
pinMode( dirUp , INPUT_PULLUP);
pinMode( dirLeft , INPUT_PULLUP);
pinMode( dirRight , INPUT_PULLUP);
pinMode( buttonRed , OUTPUT);
pinMode( buttonBlue , OUTPUT);
pinMode( buttonGreen , OUTPUT);
pinMode( buttonYellow , OUTPUT);
Serial.begin(9600);
}
void loop() {
4
```

NOTE: The USB encoder board that comes with the kit can take up to 12 buttons of input. The 4 inputs for CLR, AUTO, TURBO and MODE, are state modifiers that aren't rebindable or usable as extra button inputs.

- MODE: Switches between digital and analog mode. The joystick is not analog so this function is not useful for this build.
- TURBO: When the turbo button is held down any other button pressed will repeat rapidly.
- AUTO: sets all buttons to turbo mode until CLR is pressed.
- CLR: Clears AUTO state.

So if you connect a button to AUTO you also need a button connected to clear, otherwise turbo will be permanently turned on.

There are also 4 extra inputs for AU, AD, AL, AR connections, but

they mirror the directional inputs from the joystick ribbon cable port. Meaning you can use the ribbon cable for the joystick OR the individual connections, but not both.

// put your main code here, to run repeatedly:

```
buttonValue = analogRead(A0); //Read the analog value from pin A0
```

```
int Up = digitalRead(dirUp);
```

```
int Left = digitalRead(dirLeft);
```

```
int Right = digitalRead(dirRight);
```

```
int Down = digitalRead(dirDown);
```

```
if(Up == LOW && Left == LOW){  
Serial.println( "Direction: UPLEFT");  
}
```

```
else if(Up == LOW && Right == LOW){  
Serial.println( "Direction: UPRIGHT");  
}
```

```
else if(Down == LOW && Right == LOW){  
Serial.println( "Direction: DOWNRIGHT");  
}
```

```
else if(Down == LOW && Left == LOW){  
Serial.println( "Direction: DOWNLEFT");  
}
```

```
else if(digitalRead( dirUp ) == LOW){  
Serial.println( "Direction: UP");  
}
```

```
else if(digitalRead( dirLeft ) == LOW){  
Serial.println( "Direction: LEFT");  
}
```

```
else if(digitalRead( dirRight ) == LOW){  
Serial.println( "Direction: RIGHT");  
}
```

```
else if (digitalRead( dirDown ) == LOW){  
Serial.println( "Direction: Down");  
}
```

```
else{ //This is where you would add an action for when no  
direction is pressed.  
}
```

```
// The resistors in the circuit will cause different resistances to be  
read for each button when pressed
```

```
// This is how the Arduino knows which specific button is being  
pressed when all of the buttons are connect to a single pin
```

```
// Here is where we tell the Arduino what ranges to look for for  
each button
```

```
//for the Red button:
```

```
if (buttonValue>=1010 && buttonValue<=1015){  
  Serial.println( "RED");  
}  
//for the Blue button:  
else if (buttonValue>=1000 && buttonValue<=1008){  
  Serial.println( "BLUE");  
}  
//for the Green button:  
else if (buttonValue>=900 && buttonValue<=950){  
  Serial.println( "GREEN");  
}  
//for the Yellow button:  
else if (buttonValue>=500 && buttonValue<=600){  
  Serial.println( "YELLOW");  
}  
else{  
  // This is where you would add an action for when no  
  button is pressed.  
  delay(100);}  
}  
DIY Arcade Controller
```