

Using the Roboduino for Robots

An introduction

2009-05-16

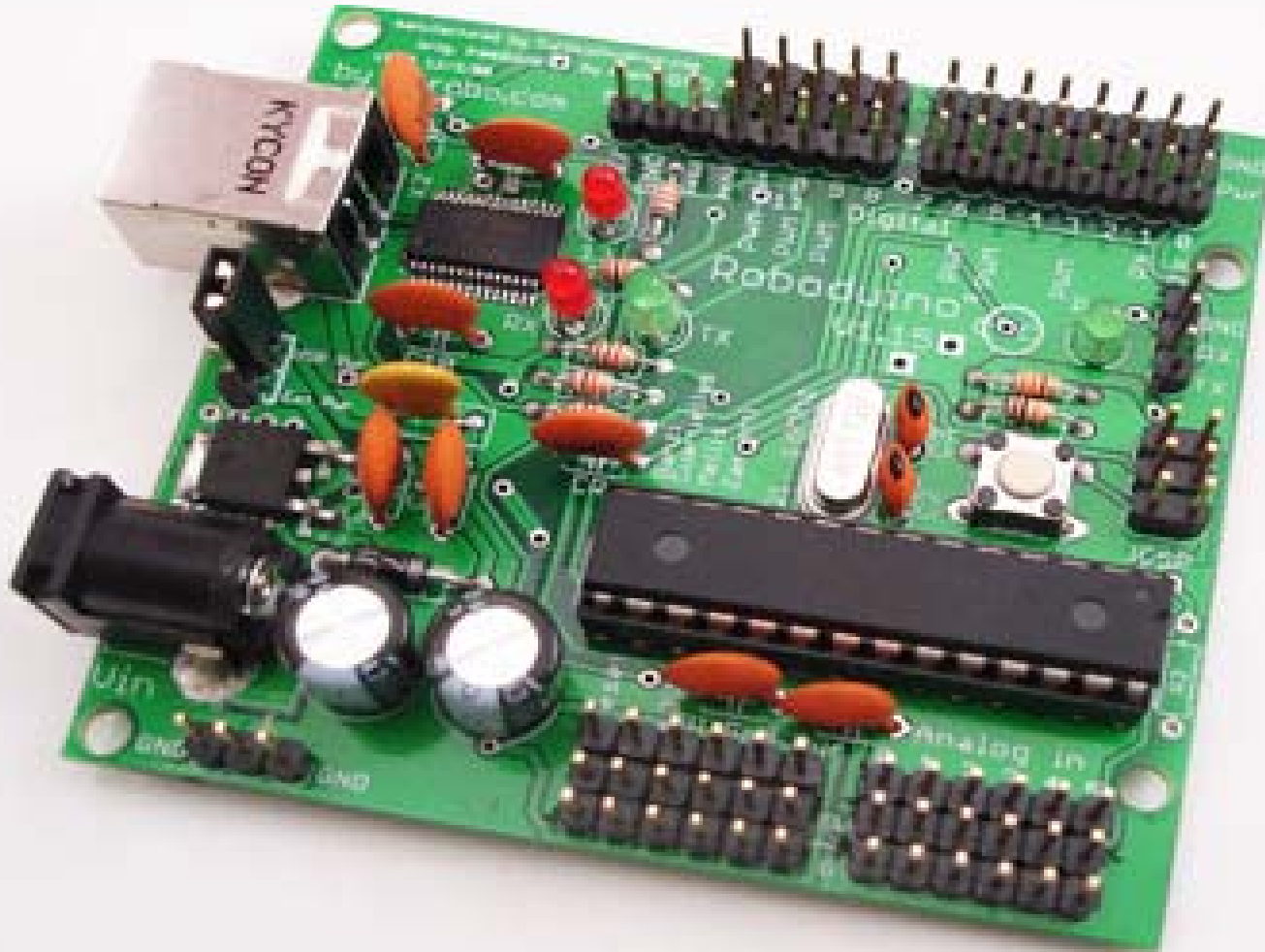
Glenn Pipe

Software Engineer/Electrical Engineer

What is a Roboduino

- An electronic board used to control one or more devices
- Can connect to many devices that are outputs
 - Servos
 - Lights
 - Motors
 - Sounds
- Can connect to many devices that are inputs
 - Switches
 - Radio receiver
 - PIR sensor, ultrasonic sensor, IR sensor, GPS, compass, position sensor, etc.

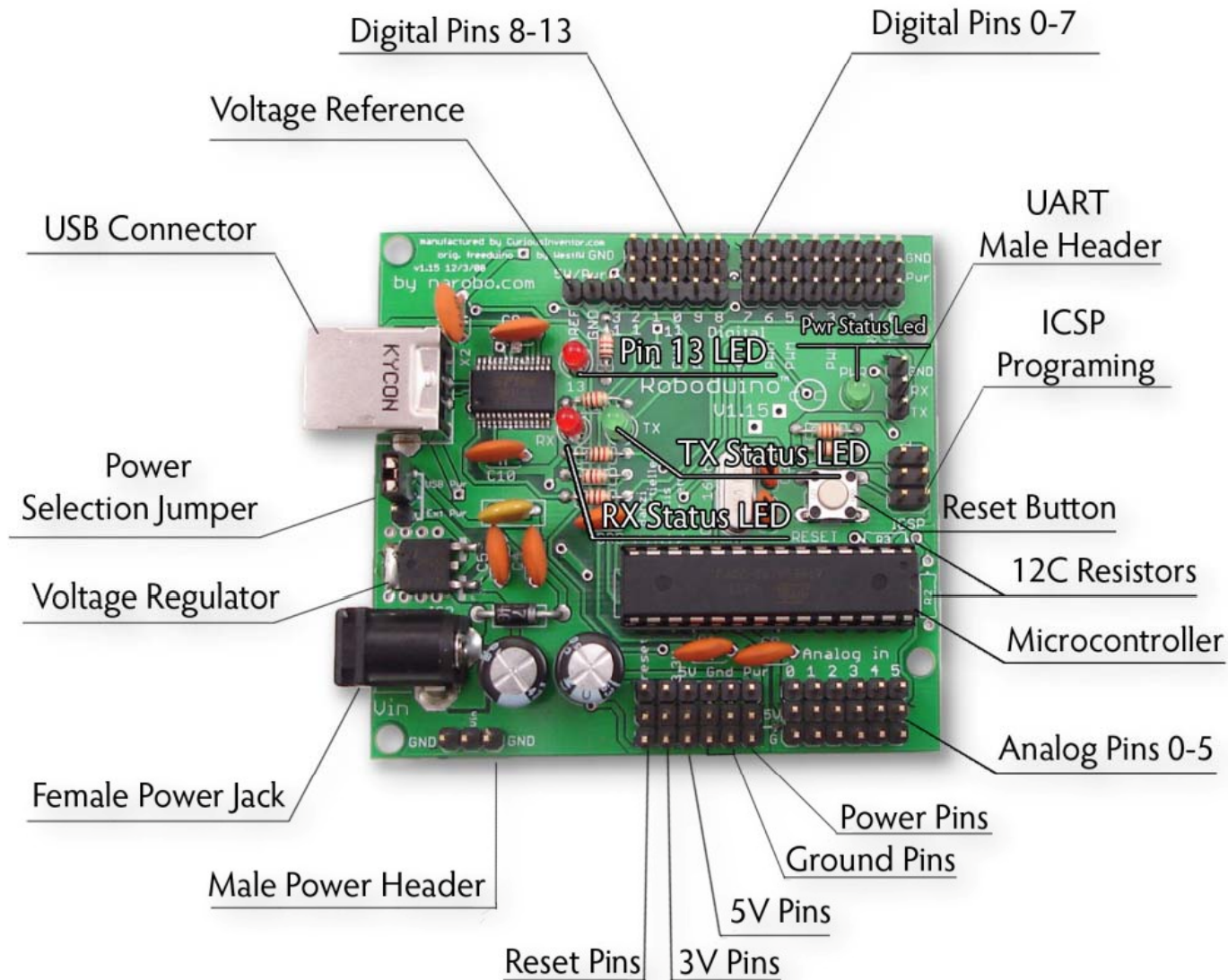
What does it look like



Why would I use a Roboduino

- Automating complex tasks with a simple trigger
 - Moving the fingers for a wave gesture for C3PO
 - Opening door and triggering fire extinguisher for R2
 - Move R2 periscope up, turn on lights, scan randomly
 - Moving torso parts on B9 for a static display
- Number of servos controlled are not limited by the number of channels on your radio
 - Draw an L with 1 radio stick to trigger a animation sequence on the robot like play the Leia message on R2 and flicker the holo projector light
 - Move several servos in different directions with 1 radio channel
- Make the robot react to the public
 - Trigger R2's dome to turn to the direction of a person when they approach and play a sound
 - Randomly play sounds at random intervals

How to connect things to the board



Elliot Lockerman: photography.narobo.com

What can connect where

- Servos can connect to any digital pin or analog pin but work best on the pins marked “PWM”
- Any pin can act as an input - this is the default mode
 - switch, radio, sensor
- Any pin can act as an output
 - light, sounds, servo, motor
- Analog pins can take an analog input
 - Analog is a voltage anywhere between 0 and 5 volts
 - Digital is only 0 or 5 volts (on or off)
- When a pin is in output mode it can supply up to 40 ma of current, which is enough to drive an LED or very low current relay
- Serial devices can be connected to RX and TX (receive and transmit)

What can connect where, part 2

- Serial devices are (use UART RX, TX pins)
 - Some ultrasonic sensors
 - GPSs
 - J.E.D.I. Display by Scott Gray
 - Vmusic2 MP3 player which generates sound
- Power can be through
 - USB connector to computer
 - Barrel jack connector
 - 0.1" spaced male pins
 - Servos can be powered separately from microcontroller or they can be powered from the same source
 - The board contains a voltage regulator for the microcontroller which will take any voltage between 6-12 volts

R2 Fire Extinguisher Example 1

- In this example I will show how to control the R2 fire extinguisher from a single switch on a R/C radio
- Without a Roboduino this would require 2 R/C switches
- Wiring
 - Connect Roboduino board to 5-6V to power servos and microcontroller
 - Connect servo for door to pin 9
 - Connect solenoid for spray to pin 10 through a Team Delta switch RCE200
 - Connect radio receiver signal pin to Roboduino pin 2

R2 Fire Extinguisher Example 2

- Desired behavior
 - When radio switch is flipped up
 - Open the door
 - Wait a quarter second
 - Fire spray solenoid
 - When radio switch is flipped down
 - Stop spray solenoid
 - Close the door
- Code for Roboduino to perform this behavior
 - See notes/handouts

R2 Fire Extinguisher Example 3

```
#include <ServoDecode.h>
#include <ServoTimer2.h>

#define doorClosed 1000
#define doorOpened 2000
#define extinguisherOff 1000
#define extinguisherOn 2000
#define switchOff 1250
#define switchOn 1750
#define doorServo 9
#define entinguisherSolenoid 10
#define switchChannel 5
```

```
ServoTimer2 door;
ServoTimer2 extinguisher;
```

```
char * stateStrings[] = {
    "NOT_SYNCHED", "ACQUIRING", "READY"};
```

R2 Fire Extinguisher Example 4

```
void setup()                // run once, when the sketch starts
{
  Serial.begin(9600);
  ServoDecode.begin();
  door.attach(doorServo);
  extinguisher.attach(extinguisherSolenoid);
  door.write(doorClosed); // make sure door is closed at startup
  extinguisher.write(extinguisherOff); // make sure the extinguisher is off at startup
}

void loop()                 // run over and over again
{
  int pulsewidth;
  // print the decoder state
  while ( ServoDecode.getState() != READY_state) {
    Serial.print("The decoder is ");
    Serial.println(stateStrings[ServoDecode.getState()]);
  }
}
```

R2 Fire Extinguisher Example 5

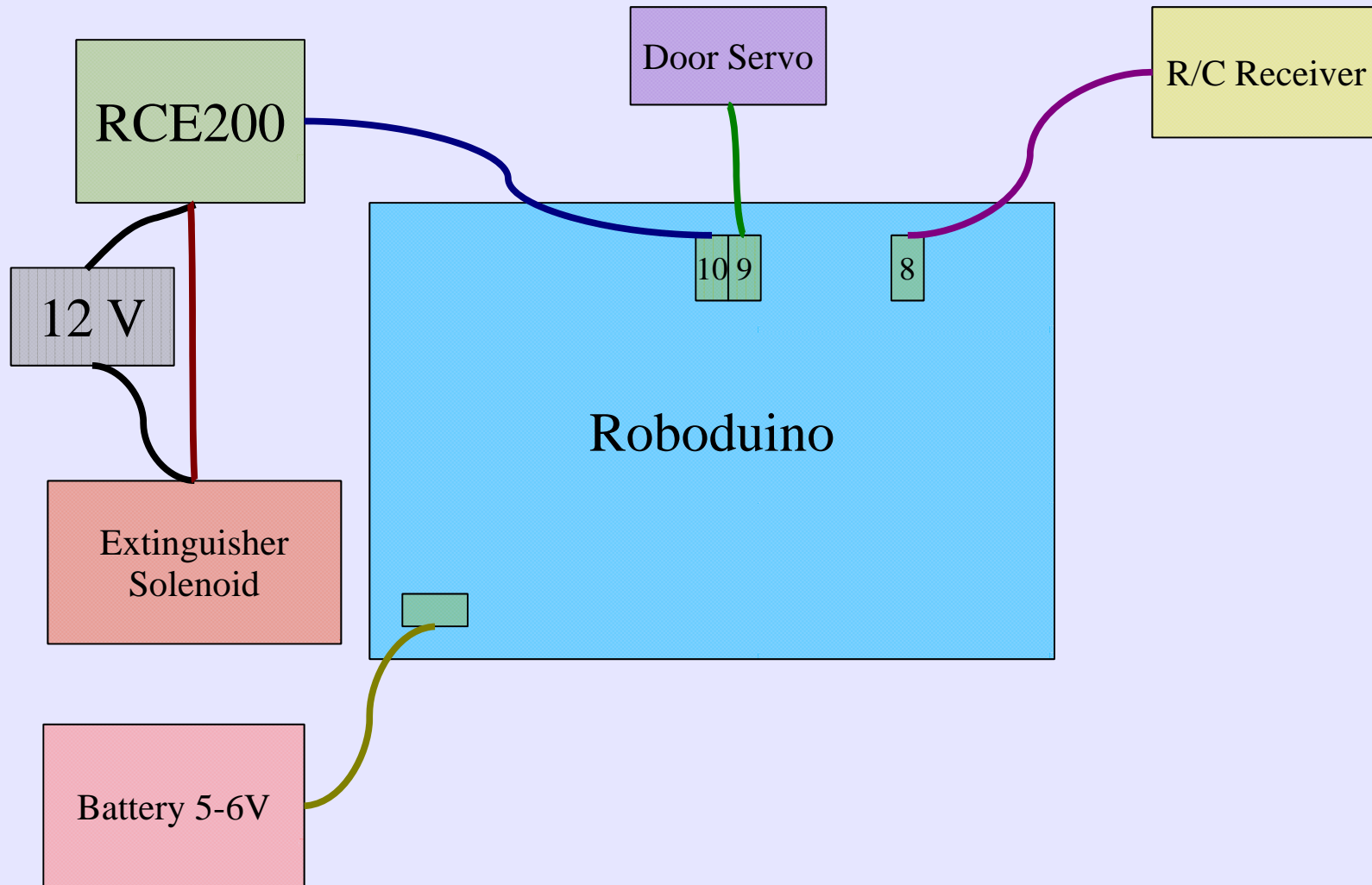
```
// decoder is ready, print the channel pulse widths
pulsewidth = ServoDecode.GetChannelPulseWidth(5);
// check if switch is on
if (ServoDecode.GetChannelPulseWidth(switchChannel) > switchOn) {
  Serial.println("Switch on");
  // if it is on then do the following:
  // 1) open the extinguisher door
  // 2) wait a quarter second
  // 3) turn on extinguisher while the switch is on
  door.write(doorOpened); // open the door
  delay(250); // wait a total of 250 ms or a quarter second
  extinguisher.write(extinguisherOn); // turn on the extinguisher
} else if (ServoDecode.GetChannelPulseWidth(switchChannel) < switchOff) {
  Serial.println("Switch off");
  // if switch is off, do the following:
  // 1) turn off the extinguisher
  // 2) close the door
  door.write(doorClosed); // close the door
  extinguisher.write(extinguisherOff); // turn off the extinguisher
}
```

R2 Fire Extinguisher Example 6

```
}  
  
delay(40); // wait 40 ms  
}
```

R2 Fire Extinguisher Example 7

Block wiring diagram



R2 Fire Extinguisher References

- Software for decoding R/C receiver signals
 - <http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1204020386/156#156>
- Wiring an R/C receiver to an arduino
 - http://paparazzi.enac.fr/wiki/Other_Hardware
- RCE200 R/C controlled switch
 - <http://www.teamdelta.com/products/prod2.htm>
- Fire extinguisher solenoid (trunk release solenoid)
 - <http://www.thetwistergroup.com/product/directed-522t-trunk-release/522T%20D26558.html>
- Fire extinguisher (air in a can)
 - http://shop1.frys.com/product/5851483?site=sr:SEARCH:MAIN_RSLT_PG
- Roboduino Store (\$40 kit, \$55 assembled)
 - http://shop1.frys.com/product/5851483?site=sr:SEARCH:MAIN_RSLT_PG