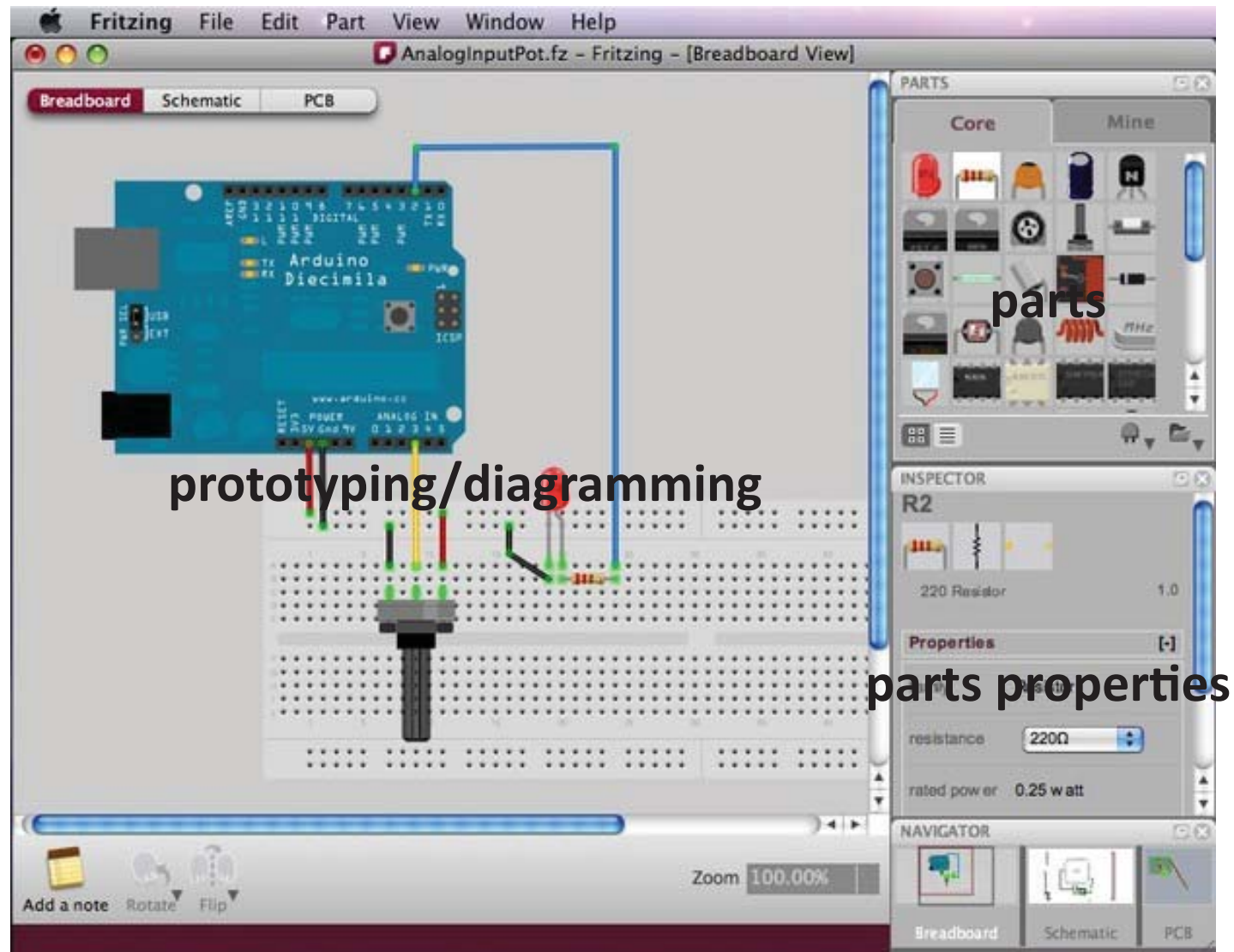


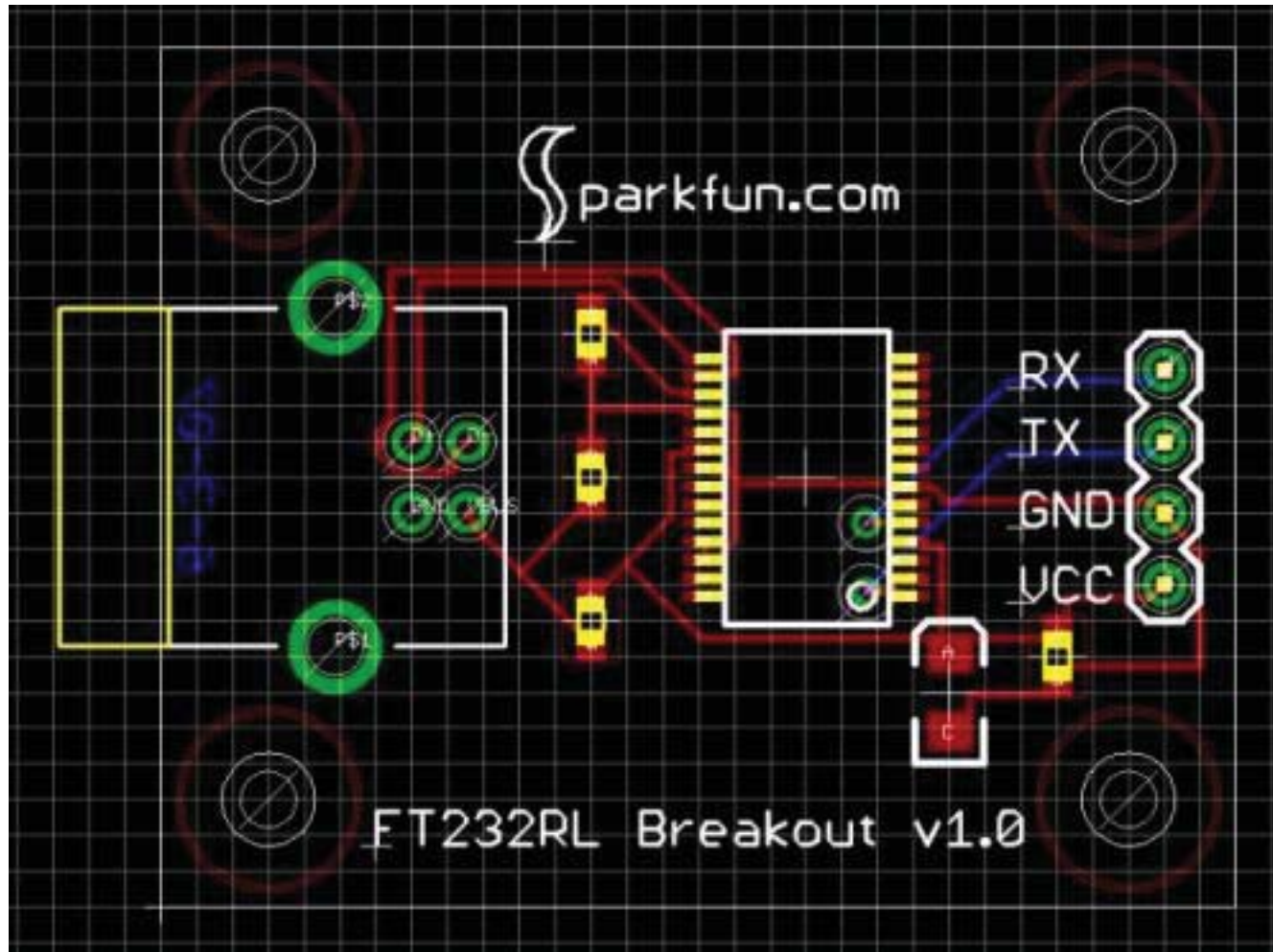
EagleCad and Fritzing_

CUSTOM PCB FABRICATION

FRITZING



EAGLECAD



External Interrupts_ARDUINO

Interrupts Introduction

When are they used?

- _ Counting pulses from an encoder (they are really short, and come very often)
- _ Catching some short pulse
- _ Using switches or digital IR sensors as bumpers (and you want an instant stop)

Why are they used?

- _ Stops the running program. ie an emergency stop.
- _ Runs the interrupt program. ie running a different program or configuration
- _ Restarts the running program. ie a reboot.

Interrupts_Arduino

Interrupts can be run if there are any changes in the designated interrupts PIN.

Atmega 168/328: 2 interrupt pins

0 interrupt pin ... corresponding to digital PIN2

1 interrupt pin ...corresponding to digital PIN3

Atmega 1250(Mega):

0 interrupt pin ... corresponding to digital PIN2

1 interrupt pin ...corresponding to digital PIN3

2 interrupt pin ... corresponding to digital PIN21

3 interrupt pin ...corresponding to digital PIN20

4 interrupt pin ... corresponding to digital PIN19

5 interrupt pin ...corresponding to digital PIN18

Interrupts_Arduino

Calling interrupt function:

```
AttachInterrupt(interrupt, function, trigger);  
detachInterrupt(interrupt);
```

Interrupt : PIN0/1 : digital 2/3

Function: function to run during interrupt

Trigger:

- LOW: when signal is low

- CHANGE: any change in signals

- RISING: goes low to high

- FALLING: goes high to low

Interrupts/Timing_Arduino

System clock works of interrupt.

So when arduino interrupt is running the function it halts system clock.

This is a problem as you cannot use time dependent functions: `delay()` , `millis()`, `serialRead/serialWrite()`

```
void setup() {  
}
```

```
void loop()  
{  
  noInterrupts();  
  // critical, time-sensitive code here  
  interrupts();  
  // other code here  
}
```

Interrupts_Arduino

```
void setup() {  
  Serial.begin(9600);  
  attachInterrupt(0, light, RISING);  
  attachInterrupt(1, readlight, CHANGE);  
}  
void loop()  
{  
  digitalWrite(13, HIGH);  
}  
void light()  
{  
  digitalWrite(13, LOW);  
  Serial.println("light off");  
}  
void readlight()  
{  
  int val = analogRead(5);  
  Serial.println(val);  
  val = map(val, 0, 1023, 300, 1800);  
  tone(5, (val*3), 600);  
}
```

Interrupts/Timing Example_Arduino

```
#include <Servo.h>
Servo myservo;
volatile int val = 0;
int pos = 0;
void setup() {
  myservo.attach(9);
  attachInterrupt(0, stopservo, CHANGE);
  attachInterrupt(1, play, CHANGE);
}
void loop()
{
  interrupts();
  for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    // in steps of 1 degree
    myservo.write(pos);          // tell servo to go to position in variable 'pos'
    delay(15);                  // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos >= 1; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);          // tell servo to go to position in variable 'pos'
    delay(15);                  // waits 15ms for the servo to reach the position
  }
  if (val == 1)
  {
    tone(13,740,300); //pin for output, freq., duration
    delay(100);
    tone(13,494,300);
    delay(200);
    tone(13,370,300);
    delay(100);
    tone(13,220,300);
    delay(200);
    tone(13,247,300);
    delay(100);
    tone(13,330,300);
    delay(200);
    tone(13,587,300);
    delay(100);
    noTone(13);
  }
  val = 0;
}

void stopservo()
{
  myservo.detach();
}

void play()
{
  val = 1;
}
```