# SIT315 – Programming Paradigms

## M1.T5HD - RIOT OS

For this task I re implemented M1.T2 via the RIOT OS. I had to make significant changes to the T2 program as it used a lot of Arduino functions but I reimplemented it with bit code.

<https://github.com/gregorymcintyre/ProgrammingParadigms.git>

The source code can be found as an .ino in the M1.T2 folder, .sketch in the M1.T5 folder. The changes between should be minimal.

A video demonstration can be found at:

<https://youtu.be/UDzzAdTPg5Y>

## Source Code

/\* Interrupt-driven Board

\* 14/3/19

\* Greg McIntyre

\*

\* Bitwise interrupt implementation of T1 for T5

\*

\* https://github.com/gregorymcintyre/ProgrammingParadigms.git

\*

\*/

#define PIR 2

#define LED\_PIN 5

int input = 0;

volatile bool LEDstate = false;

ISR(INT0\_vect)

{

//LEDstate = !LEDstate;

//digitalWrite(LED\_BUILTIN, LEDstate);

PORTD ^= (1 << LED\_PIN);

//Serial.println("Interrupt has occured");

puts("Interrupt has occured");

}

void setup() {

//Serial.begin(9600);

//Serial.println("Program Running...");

puts("Interrupt Program Running...");

//pinMode(LED\_BUILTIN, OUTPUT); //built in LED

DDRD |= (1<<LED\_PIN);

//pinMode(PIR, INPUT);

DDRD &= ~(1 << PIR); //PIR Input

//attachInterrupt(digitalPinToInterrupt(PIR), change, CHANGE);

//CHANGE in state

EICRA |= (1 << ISC00);

EICRA &= ~(1 << ISC01);

//ENABLE interrupts on INT0

EIMSK |= (1 << INT0);

//set Global interrupts

sei();

}

void loop() {

delay(1000);

}