int main(void) {

// Configure the A/D module and port RB4 as analog input

TRISBbits.TRISB4 = 1;

AD1PCFGbits.PCFG4= 0;

AD1CON1bits.SSRC = 7;

AD1CON1bits.CLRASAM = 1;

AD1CON3bits.SAMC = 16;

AD1CON2bits.SMPI = 2-1;

AD1CHSbits.CH0SA = 4;

AD1CON1bits.ON = 1;

while(1)

{

AD1CON1bits.ASAM = 1;

while( IFS1bits.AD1IF == 0 );

printInt(ADC1BUF0, 16 | 3 << 16);

putChar('\n');

int \*p = (int \*)(&ADC1BUF0);

int media = 0;

for(; p <= (int \*)(&ADC1BUFF); p+=4 ) {

media += \*p;

}

media /= SAMPLES;

V = (media \* 33 + 511) / 1023;

// Reset AD1IF

IFS1bits.AD1IF = 0;

}

return 0;

}

// configure RB8-RB14 as outputs

// configure RD5-RD6 as outputs

TRISB &= 0x80FF;

TRISD &= 0xFF9F;

void send2displays(unsigned char value)

{

static const char disp7Scodes[16] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71};

static char displayFlag = 0;

int digit\_low = toBcd(value) & 0x0F;

int digit\_high = toBcd(value) >> 4;

if (displayFlag == 0)

{

LATD = (LATD & 0xFF9F) | 0x0020;

LATB = (LATB & 0x80FF) | (disp7Scodes[digit\_low] << 8);

} else {

LATD = (LATD & 0xFF9F) | 0x0040;

LATB = (LATB & 0x80FF) | (disp7Scodes[digit\_high] << 8);

}

displayFlag = !displayFlag;

}

li $v0, RESET\_CORE\_TIMER

syscall

wait:

li $v0, READ\_CORE\_TIMER

syscall

blt $v0, 200000, wait # e.g. f = 100Hz

void delay(unsigned int ms) {

resetCoreTimer();

while (readCoreTimer() < 20000 \* ms);

}

unsigned char toBcd(unsigned char value)

{

return ((value / 10) << 4) + (value % 10);

}

int freq = 1 + (ADC1BUF0 \* 4) / 1023; // Mapeia [0, 1023] para [1, 5] Hz

int delayMs = 1000 / freq; // Calcula o delay em ms para a frequência

# Duas frequencias diferentes

# Contador com 5Hz e display com 50Hz

while(1)

{

i = 0;

do

{

send2displays(counter);

// wait 20 ms (1/50Hz)

delay(20);

} while(++i < 10); // 10 \* 20 ms = 200 ms (1/5Hz)

counter = (counter + 1) % 256;

}

if (displayCounter++ >= (freq / 50)) { //50Hz com frequencia variavel

send2displays(print\_value);

displayCounter = 0; // Resetar o contador do display

}