Microcontroladores

Semana 12 Teoría

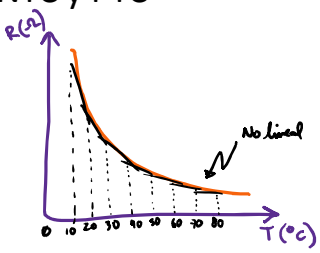
1

Sección de Preguntas

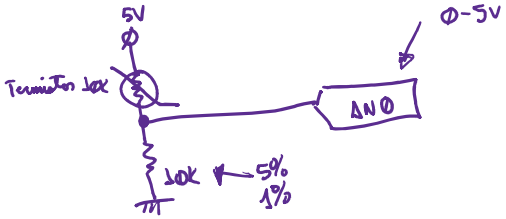
2

Termistor

- Dos tipos: NTC y PTC



0-10 → Ec Recta 1
10-20 → Ec Recta 2

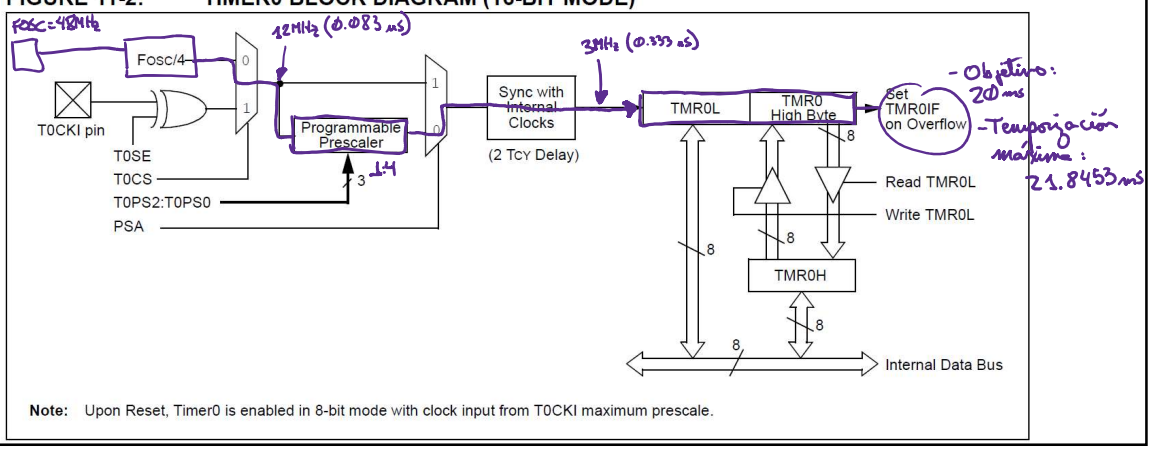


3

Uso del Timer0 en modo 16 bits para manipular un servomecanismo

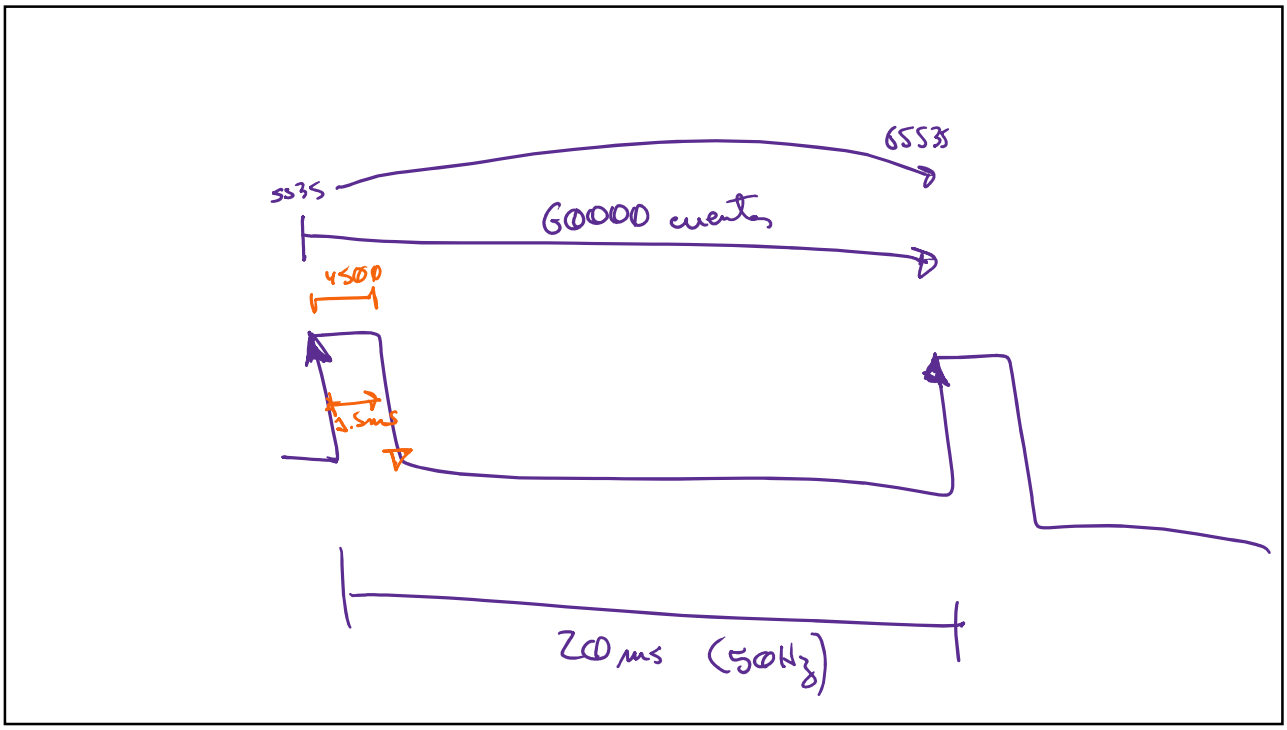
- Definir la base de tiempos para el servomecanismo: 20 ms de periodo (50Hz)

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)

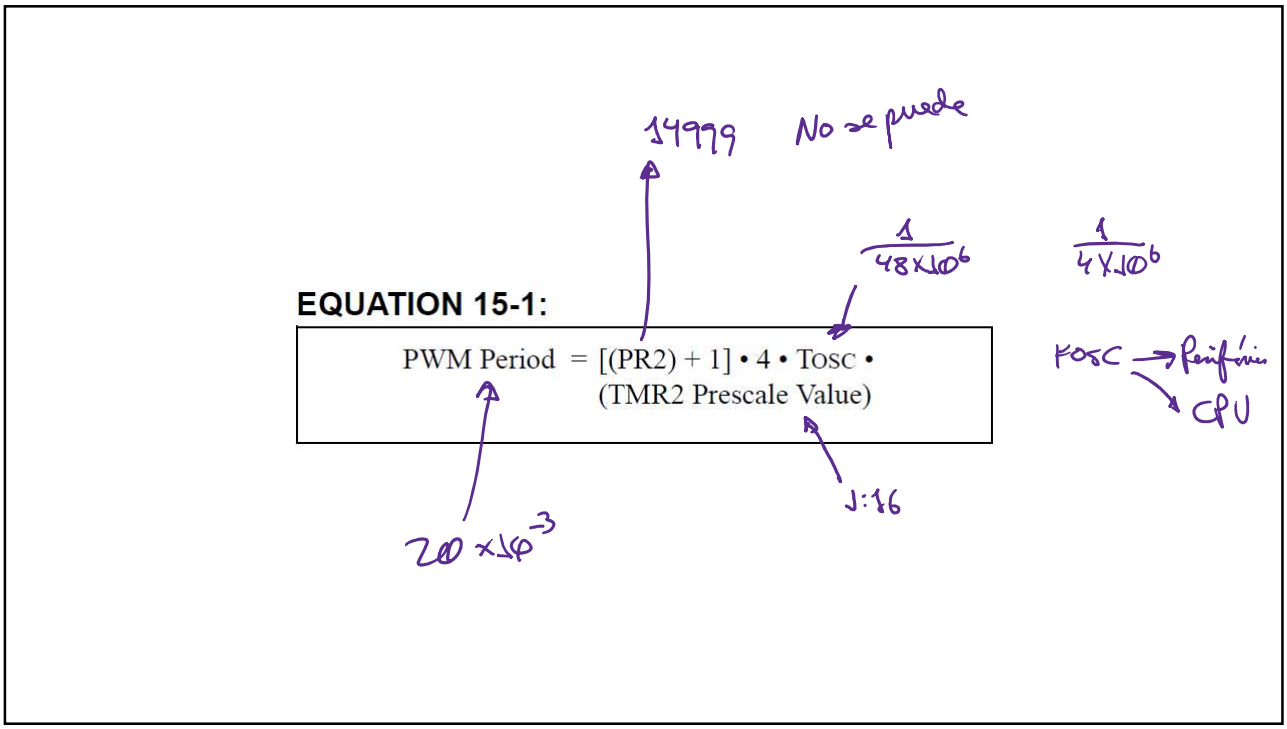


Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

4


5535
65535
60000 cuentas
4500
1.5ms
20 ms (50Hz)

5


14999
No se puede
1/48x10^6
1/4x10^6
EQUATION 15-1:
PWM Period = [(PR2) + 1] • 4 • Tosc • (TMR2 Prescale Value)
Fosc → Periférico
CPU
20 x10^-3
1:16

6

Uso del Timer0 en modo 16 bits para manipular un servomecanismo

- Prueba de Timer0 a 16bits, FOSC/4, PSC 1:4 y sin cuenta inicial (FOSC=48MHz):

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

T0CON=0X80

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT:** Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

Uso del Timer0 en modo 16 bits para manipular un servomecanismo

- Interrupciones habilitadas para Timer0

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

INTCON = 0XA0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1)
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

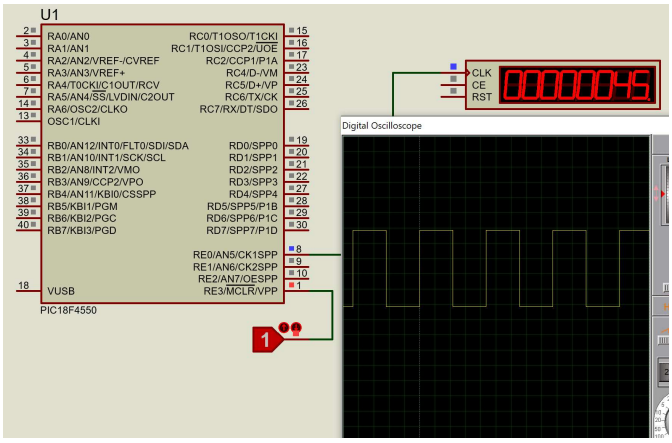
bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Uso del Timer0 en modo 16 bits para manipular un servomecanismo

- Prueba de Timer0 a 16bits, FOSC/4, PSC 1:4 y sin cuenta inicial (FOSC=48MHz):

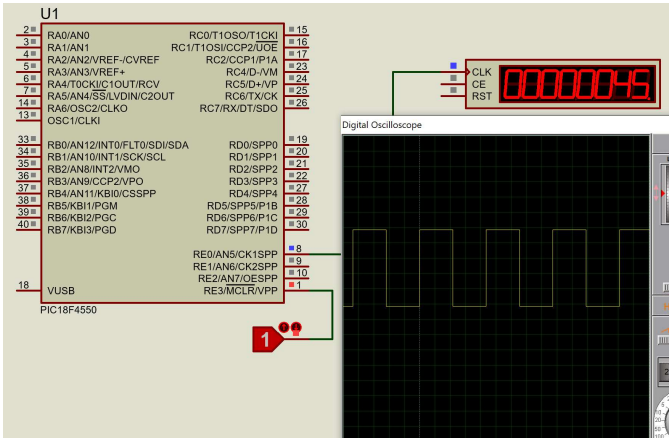
```c
#include "cabecera.h"
#include <xc.h>
#define _XTAL_FREQ 48000000UL

void init_conf(void){
    T0CON = 0x80;           //FOSC/4, PSC1:4, 16bit
    INTCON = 0xA0;          //GIE=1, TMR0IE=1
    ADCON1 = 0x0F;          //All digital I/O
    TRISEbits.RE0 = 0;      //RE0 an output
}

void main(void) {
    init_conf();
    while(1);
}

void __interrupt() TMR0_ISR(void){
    INTCONbits.TMR0IF = 0;
    asm("btg LATE, 0");
}
```



Uso del Timer0 en modo 16 bits para manipular un servomecanismo

- Prueba de Timer0 a 16bits, FOSC/4, PSC 1:4 y sin cuenta inicial (FOSC=48MHz):

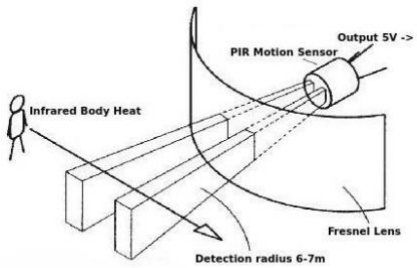

```c
#include "cabecera.h"
#include <xc.h>
#define _XTAL_FREQ 48000000UL

void init_conf(void){
    T0CON = 0x80;           //FOSC/4, PSC1:4, 16bit
    INTCON = 0xA0;          //GIE=1, TMR0IE=1
    ADCON1 = 0x0F;          //All digital I/O
    TRISEbits.RE0 = 0;      //RE0 an output
}

void main(void) {
    init_conf();
    while(1);
}

void __interrupt() TMR0_ISR(void){
    INTCONbits.TMR0IF = 0;
    asm("btg LATE, 0");
}
```



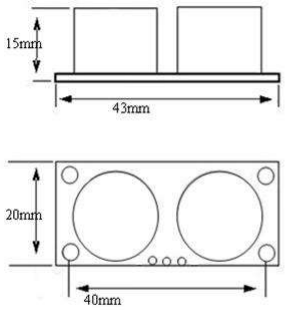
Uso del Timer0 en modo 16 bits para manipular un servomecanismo

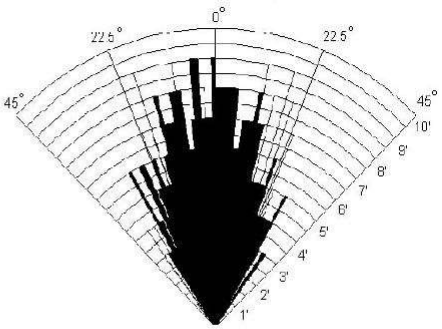
- Definir la base de tiempos para el servomecanismo: 20 ms de periodo (50Hz)

Sensor PIR



Sensor ultrasónico

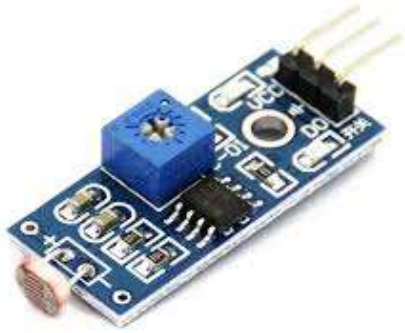




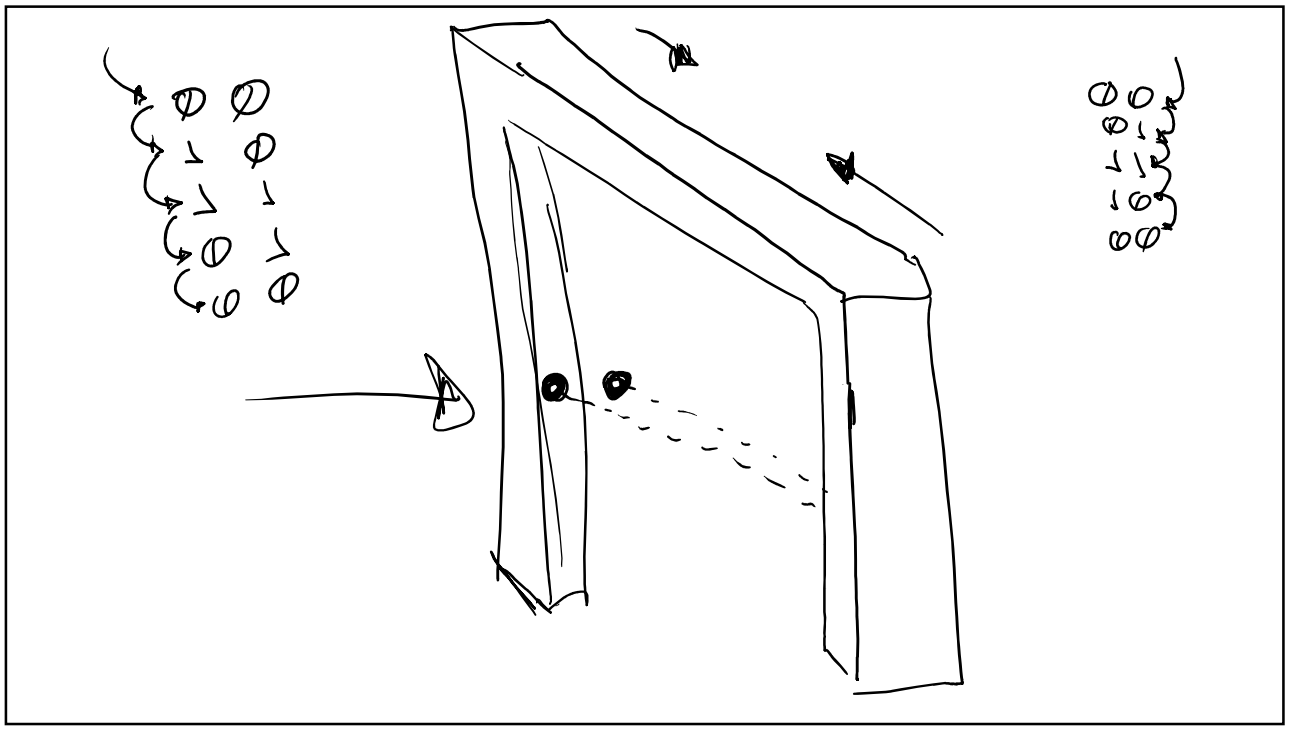
Practical test of performance,
Best in 30 degree angle

13

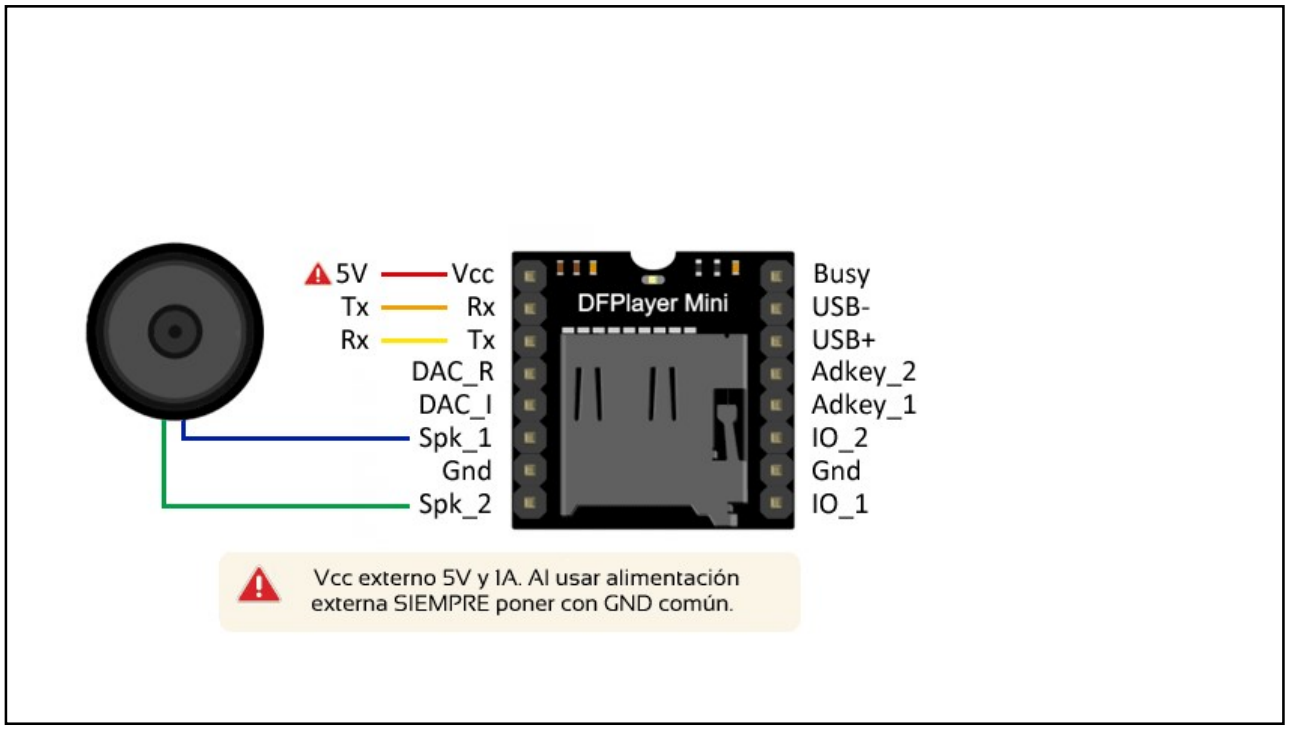
Láser con LDR



14

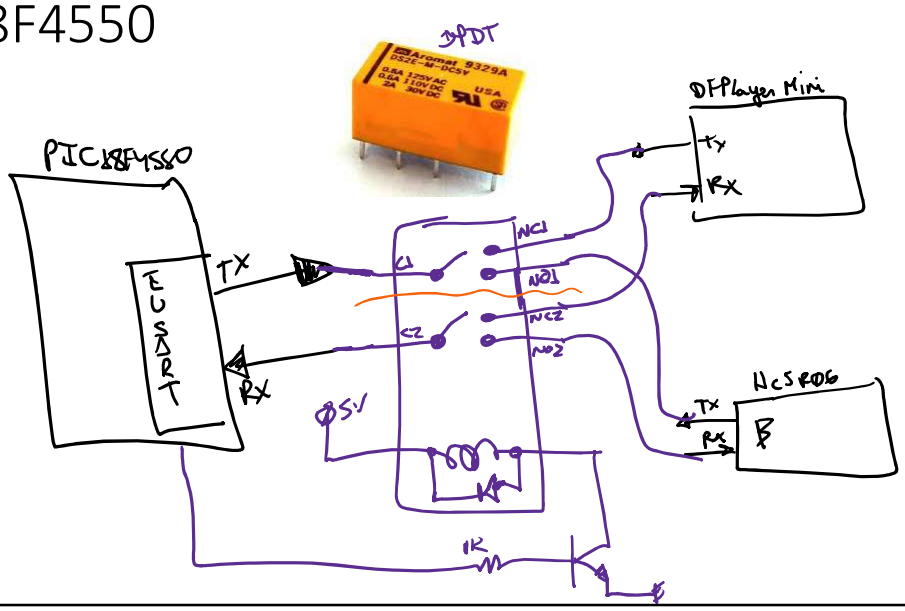


15

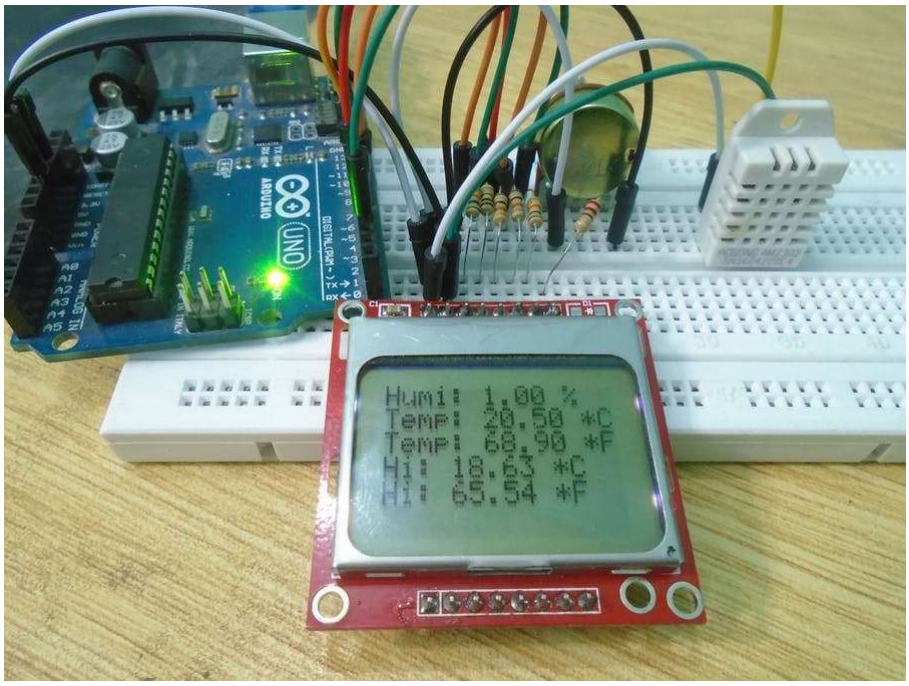

5V
Tx
Rx
Vcc
Rx
Tx
DAC_R
DAC_I
Spk_1
Gnd
Spk_2
DFPlayer Mini
Busy
USB-
USB+
Adkey_2
Adkey_1
IO_2
Gnd
IO_1
Vcc externo 5V y IA. Al usar alimentación externa SIEMPRE poner con GND común.

16

Cómo conectar dos periféricos serials al PIC18F4550



17



18