

UNIVERSIDAD CATÓLICA DE VALPARAÍSO
LABORATORIO DE ELECTRÓNICA APLICADA EIE-594

COMUNICACIÓN I2C Y EL RELOJ TIEMPO REAL

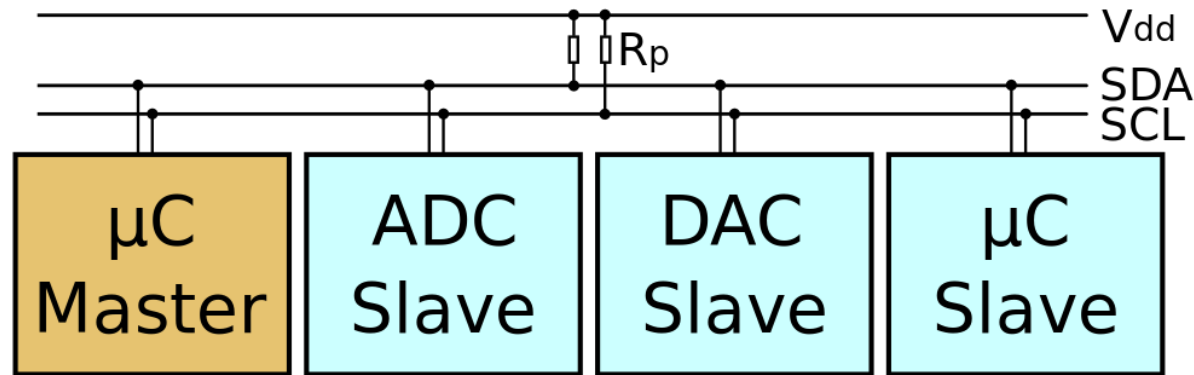
Docente: Rodrigo Olguín

Integrantes: Matías Castillo - Adrián Gallardo - Francisco Vega

¿Qué es I2C?

I2C ó IIC (Inter-Integrated Circuit), es un protocolo de comunicación serie sincronico diseñado por Philips.

Uno de sus usos mas comunes es la comunicación entre un microcontrolador y sensores periféricos



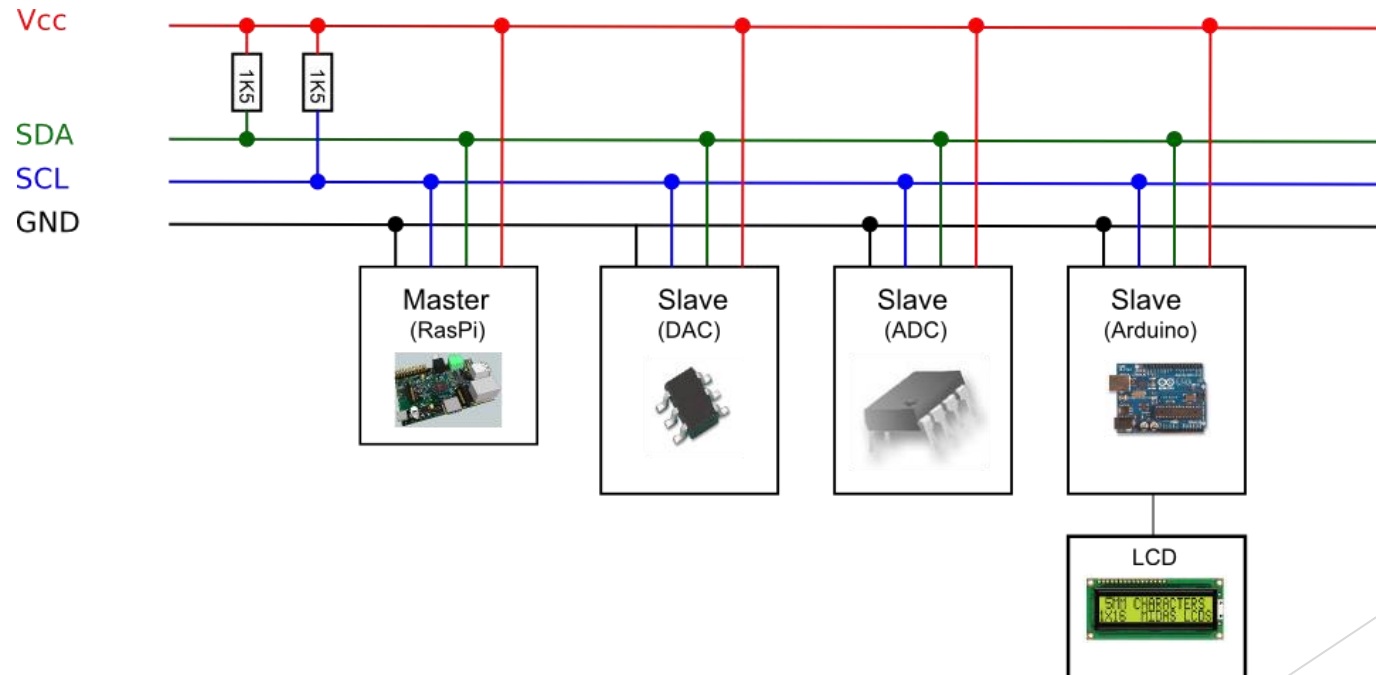
Características del protocolo

- El tipo de comunicación es half-dúplex
- Bus multi-maestro
- Velocidad standard de 100Kbit/s. Se puede cambiar al modo rápido(400Kbit/s) o modo Alta Velocidad (3.4M bits/s)
- Solo se necesitan dos líneas:
 - ▶ SDA (Serial Data Line): Línea de datos.
 - ▶ SCL (Serial Clock Line): Línea de reloj.
- Los comunicación siempre tiene la estructura siguiente:
 - ▶ Transmisor: Byte de datos (8 Bits)
 - ▶ Receptor: Bit llamado ACK de confirmación.
- En I2C cada dispositivo tiene una dirección de 7 bits, es decir se pueden tener hasta teóricamente 128 dispositivos (112 direcciones para dispositivos el resto reservadas)

Hardware

El bus I2C cuenta con tres líneas: SDA(serial data), SCL(serial clock) y Ground.

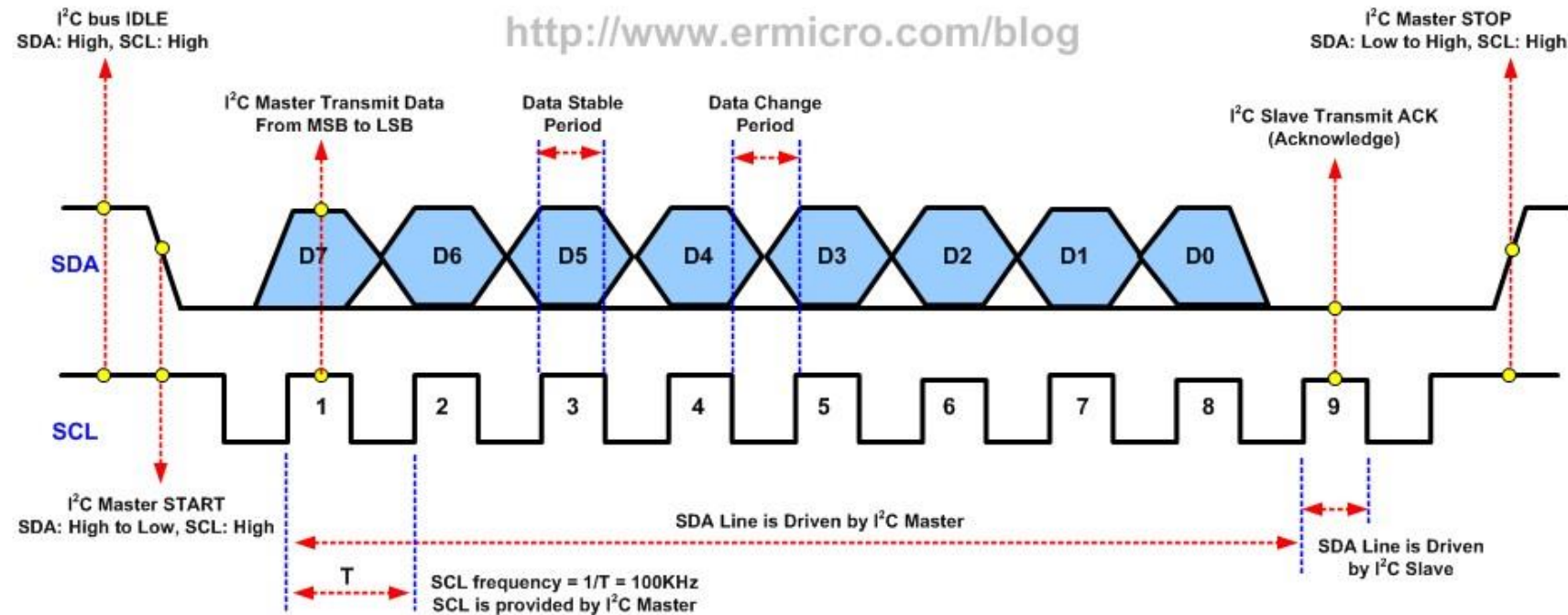
- ▶ SDA y SCL van a su pin correspondiente en cada dispositivo, de manera que todos quedan en paralelo.
- ▶ Las líneas SDA y SCL están independientemente conectadas a dos resistores Pull-Up que se encargaran de que el valor lógico siempre sea alto a no ser que un dispositivo lo ponga a valor lógico bajo.



Comunicación

► Transmisión de bits

Los bits de datos se transmiten por SDA y para cada bit es necesario un pulso del SCL,



I²C Bus Master and Slave Timing Diagram

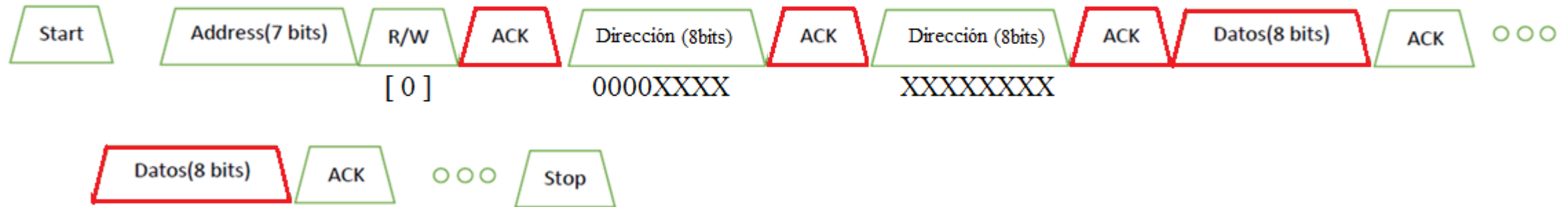
Estructura de Comunicación

1. START condition (*Master*)
2. 7 Bits de dirección de esclavo (*Master*)
3. 1 Bit de Read/Write, 0 es Leer y 1 Escribir. (*Master*)
4. 1 Bit de Acknowledge (*Slave*)
5. Byte de dirección de memoria (*Master*)
6. 1 Bit de Acknowledge (*Slave/Master (Escritura/Lectura)*)
7. Byte de datos (*Master/Slave (Escritura/Lectura)*)
8. 1 Bit de Acknowledge, (*Slave/Master (Escritura/Lectura)*)
9. *Se repite el paso 7 tantas veces como sea necesario*
10. STOP condition (*Master*)

Trama I2C



LECTURA MEMORIA ROM 4K



Ventajas

- ▶ La sencillez y la flexibilidad son características clave que hacen que este bus sea atractivo para muchas aplicaciones.
- ▶ Sólo se requieren dos líneas de bus
- ▶ No hay requisitos estrictos de velocidad de transmisión, como por ejemplo con el RS232, el maestro genera un reloj de bus.
- ▶ Mecanismo para asegurar que el mensaje ha llegado a su destino mediante ACK.
- ▶ Cada nodo posee una dirección con el cual podemos comunicarnos individualmente

Desventajas

- ▶ No es Full-Duplex por lo que no permite envíos y recepciones al mismo tiempo.
- ▶ Velocidad no tan elevada como SPI
- ▶ En el modo estándar número de nodos limitado a 112.
- ▶ No posee comprobación de datos

DS1307

RTC (Real Time Clock)

Reloj de tiempo real para los amigos

Descripción General del dispositivo

- ▶ RTC de bajo consumo
- ▶ Se comunica por I2C con codificación decimal binaria (BCD)
- ▶ Entrega segundos, minutos, horas, día, fecha, mes y año (incluyendo correcciones de meses y años bisiestos hasta el año 2100).
- ▶ Compatible con formato de 24 hr y 12 hr AM/PM
- ▶ Respaldo de suministro eléctrico
- ▶ Salida con señal cuadrada programable

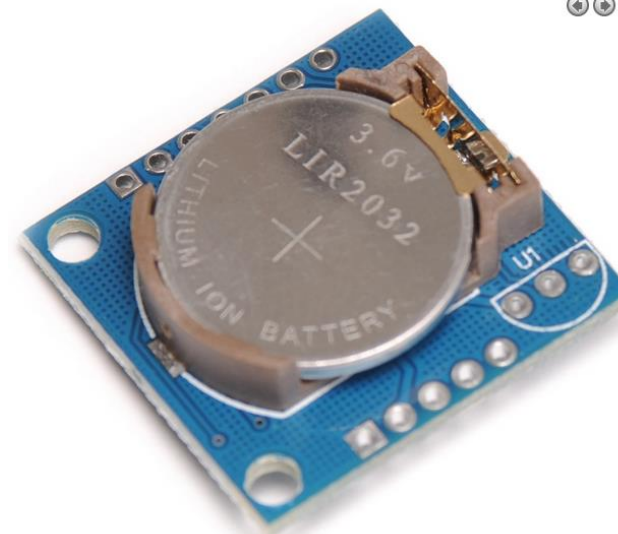
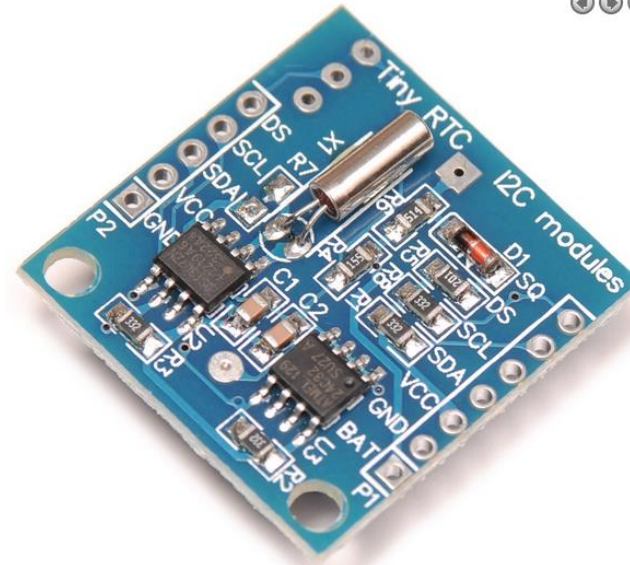
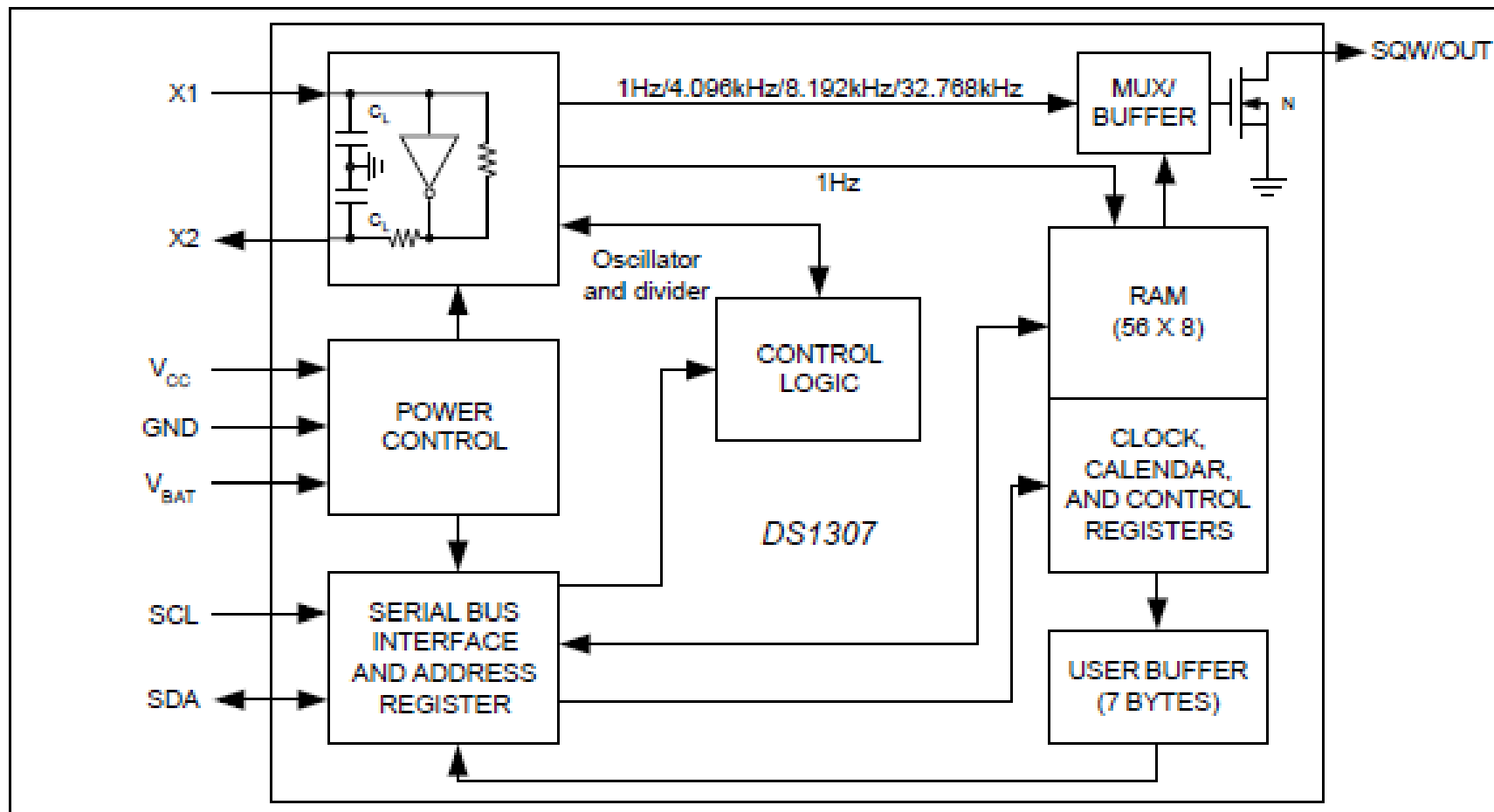
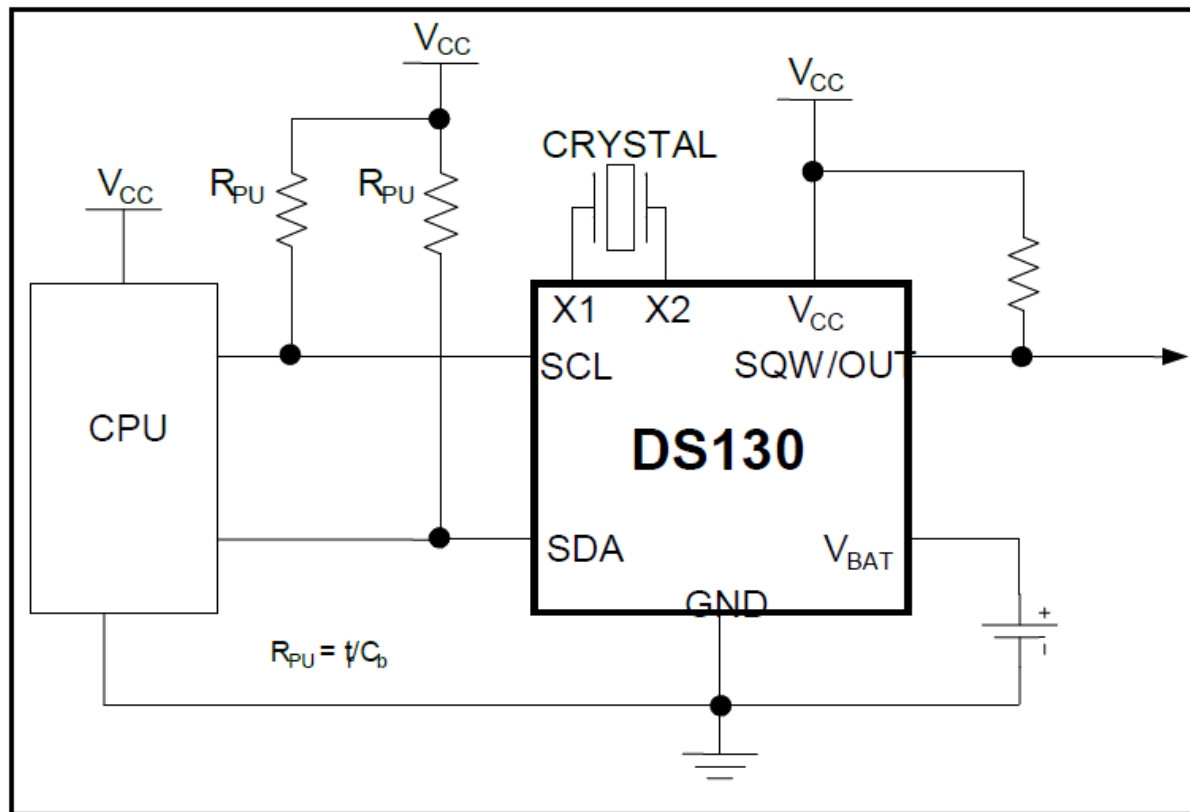


Diagrama de bloques DS1307



Circuito DS1307



Registros DS1307

► Registro de cronómetro

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

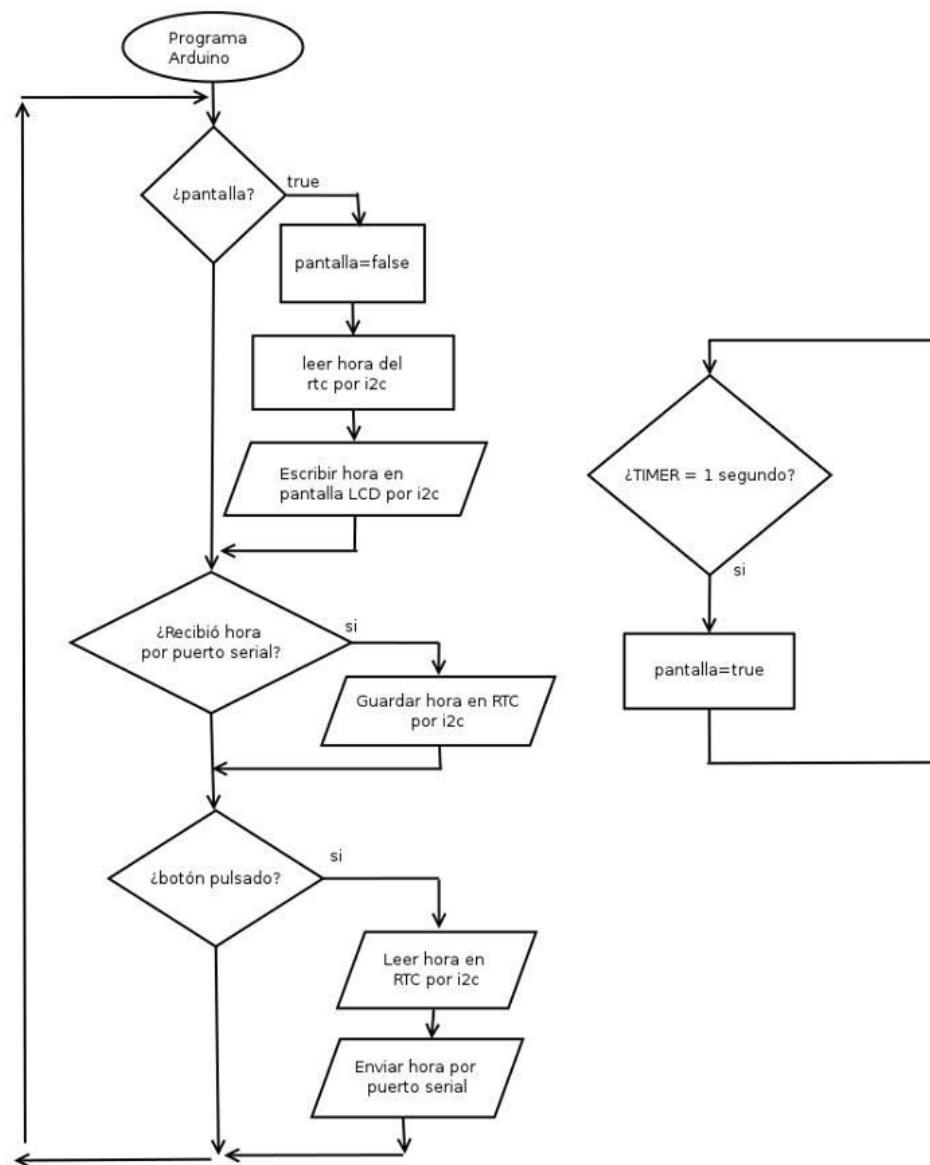
0 = Always reads back as 0.

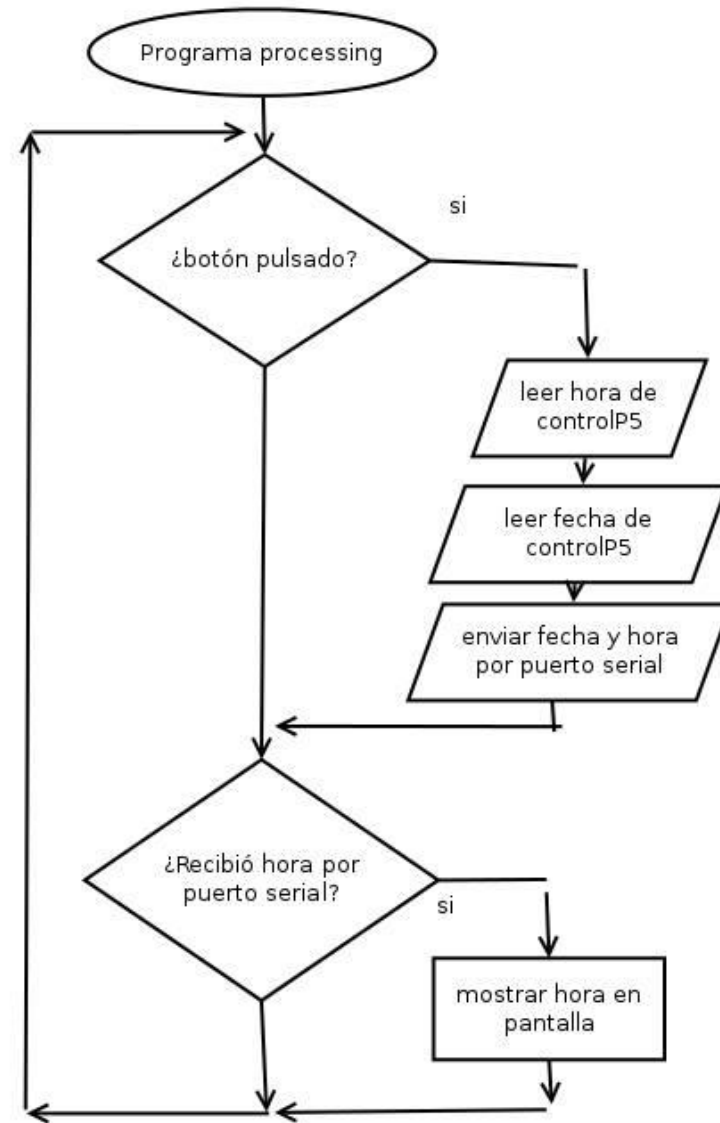
Registros DS1307

► Registro de control

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1





Arduino:

Librerías: RTCLib.h

Wire.h

RTC_DS1307 RTC;

setup():

Wire.begin();

RTC.begin();

if (!RTC.isrunning()) -> RTC.adjust(DateTime(__DATE__, __TIME__));

processing quiere ajustar la hora:

RTC.adjust(DateTime(sFecha, sHora));

boton pulsado, enviarle hora a processing:

DateTime now = RTC.now();

now.year()

now.month()

now.day()

now.hour()

now.minute()

now.second()