BARSicle - the learn, code, build project

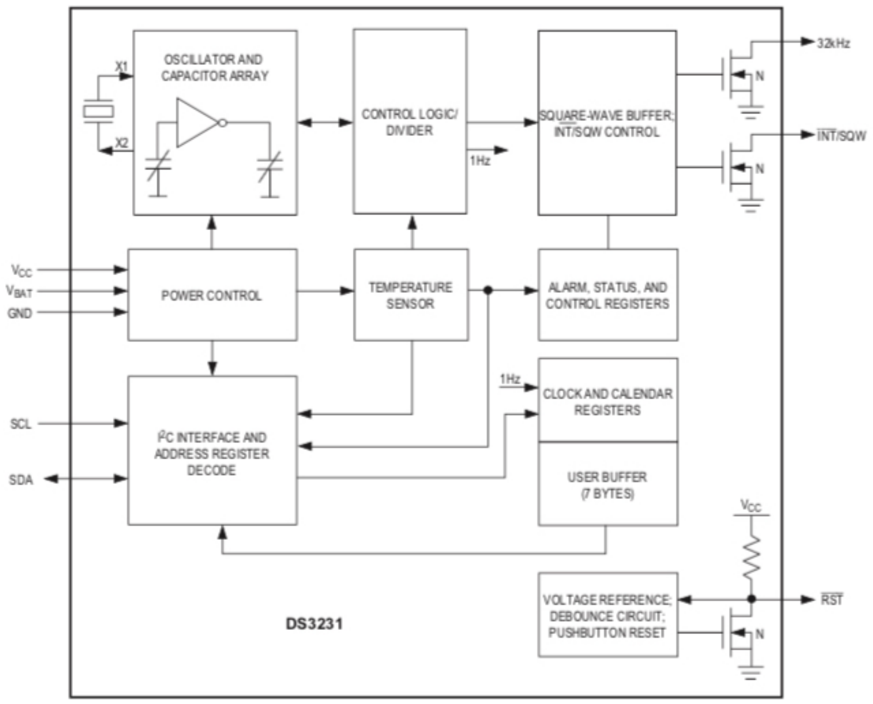
The BARScile project is a beginners learning, coding and building project of the Banbury Amateur Radio Society. It will cover,

* the use of the Arduino Nano
* the build of a digital Signal Generator / VFO, an RF Volt/power meter, a Direct Conversion Receiver.

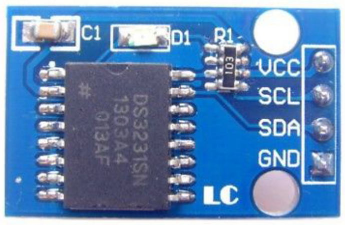
An extension of the project will cover an SSB Exciter, Power Amplifier with Low Pass Filter and an SWR/Antenna Tuning Unit

3. ARDUINO
RTC & OLED

There are some very clever ICs available, the DS3231 is just one of them. This is a real time clock. It has a built in xtal for timing, and uses/charges an external battery. A bit like the watch on your wrist.


OSCILLATOR AND CAPACITOR ARRAY
X1
X2
CONTROL LOGIC/ DIVIDER
1Hz
SQUARE-WAVE BUFFER; INT/SQW CONTROL
32kHz
INT/SQW
N
Vcc
VBAT
GND
POWER CONTROL
TEMPERATURE SENSOR
ALARM, STATUS, AND CONTROL REGISTERS
SCL
SDA
I²C INTERFACE AND ADDRESS REGISTER DECODE
CLOCK AND CALENDAR REGISTERS
USER BUFFER (7 BYTES)
VOLTAGE REFERENCE; DEBOUNCE CIRCUIT; PUSHBUTTON RESET
RST
DS3231

This is the module we are using, carrying a DS3231



The module has an I2C serial bus interface, and has a unique bus address. So it can be wired in parallel with your OLED to the four lines VCC SCL SDA & GND.

BCD
The RTC talks in Binary Coded Decimal. This is a coding where one byte of 8 bits is divided into two nibbles, and each 4 bits represents a binary number 0-15. Now we normally use decimal numbers, so we need some functions to convert backwards and forwards.

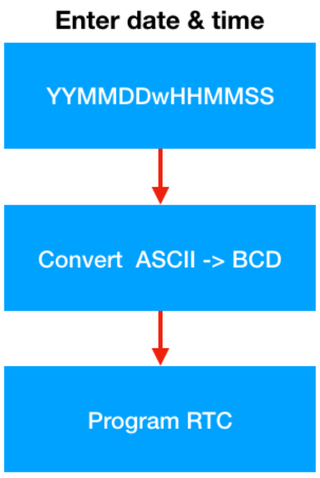
```
byte decToBcd(byte val)
{
  return( (val/10*16) + (val%10) );   // decimal - > BCD
}

byte bcdToDec(byte val)
{
  return( (val/16*10) + (val%16) );   // BCD -> decimal
}
```

Have a think about these so you understand how they work. Getting grips with various computer number formats is key skill.

Inputting the time
Plug in your CR1220 battery, +ve side visible, and mount the RTC on your breadboard. Then wire it up in parallel with the OLED four lines VCC, SCL, SDA & GND.

When you first plug in your battery the RTC does not know the date & time. So you have to set it. This is done in three steps, entering the data on the monitor window.



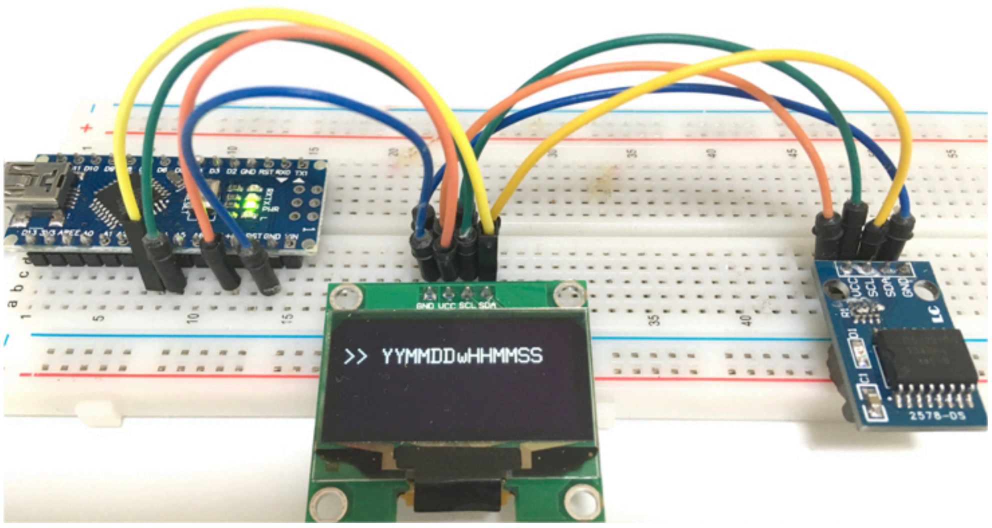
The input is in ASCII. This converted to decimal bytes,

```
void asciiToByte() {
  // convert ASCII rtc buffer string to bytes
  yr = ((byte)rtcBuf[0] - 48) * 10 + (byte)rtcBuf[1] - 48;
  mth = ((byte)rtcBuf[2] - 48) * 10 + (byte)rtcBuf[3] - 48;
  dy = ((byte)rtcBuf[4] - 48) * 10 + (byte)rtcBuf[5] - 48;
  dow = ((byte)rtcBuf[6] - 48);
  hrs = ((byte)rtcBuf[7] - 48) * 10 + (byte)rtcBuf[8] - 48;
  mns = ((byte)rtcBuf[9] - 48) * 10 + (byte)rtcBuf[10] - 48;
  sec = ((byte)rtcBuf[11] - 48) * 10 + (byte)rtcBuf[12] - 48;
}
```

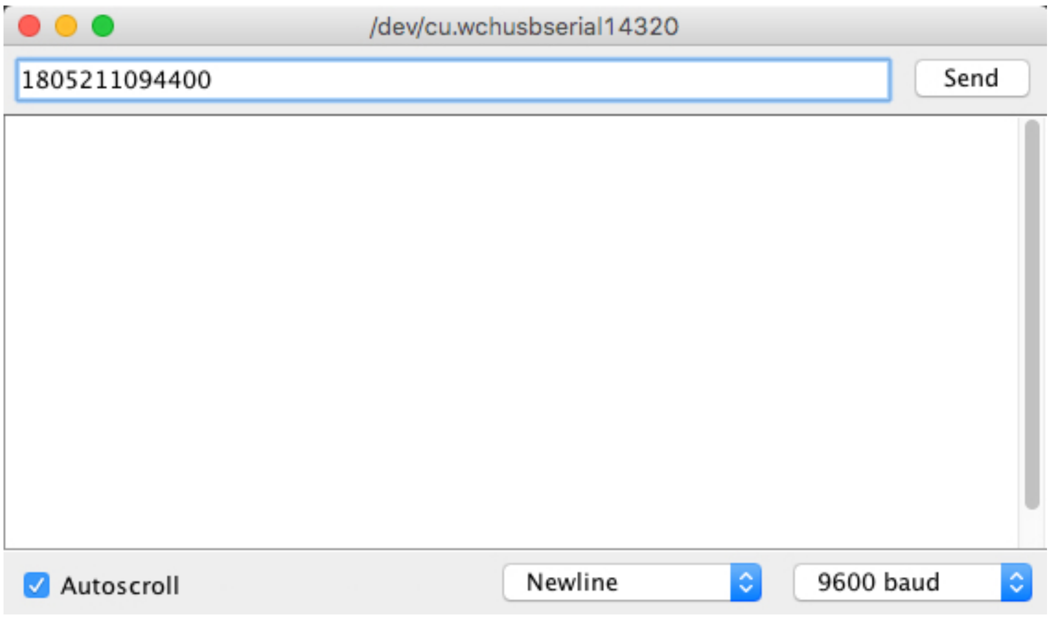
48 is the ASCII code for the number `"0"` so take one ASCII character from the string input, subtract 48 and x10 and you get the 10's as the first byte, take the second ASCII character and add it.

Setting the RTC
Open the sketch *File > Sketchbook > My_SET_RTC_OLED.* This will prompt you to enter the date and time as an character string.

"YYMMDDwHHMMSS". Year, month, date, day-of-the-week, hour, minute, second.



Open the Monitor window and enter a time in the near future, say the next minute (seconds = 00). The exactly on this time hit Send. You can use internet time or your PC if it is synchronised.



```
1805211094400
```

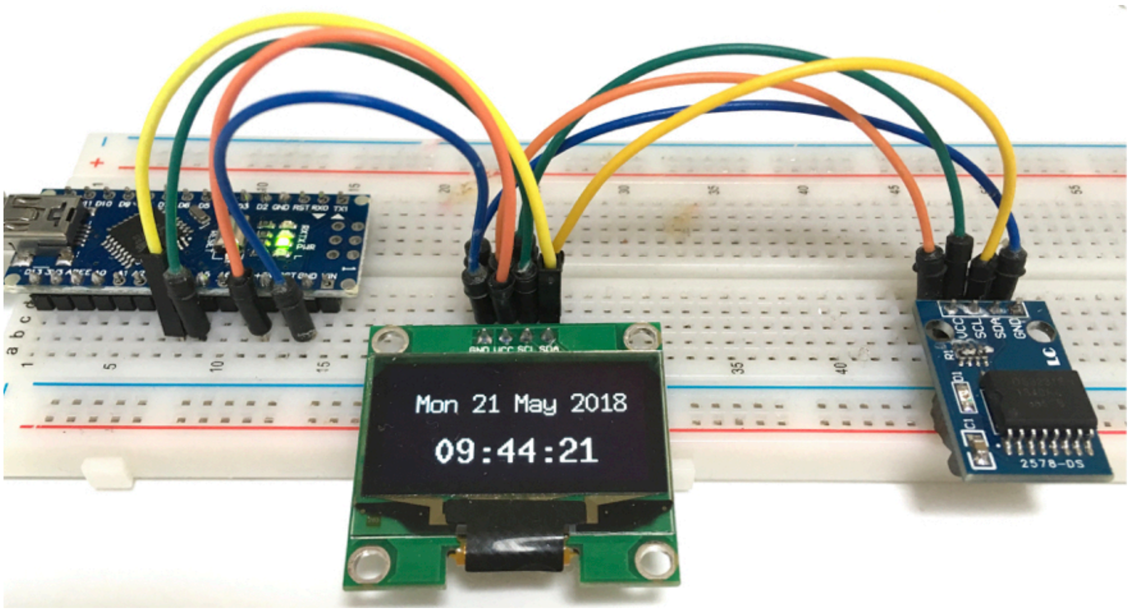
Autoscroll [✓] Newline 9600 baud

2018, May, 21st, Monday, 9am, 44min, 00sec.

The date & time are converted to bytes (see above) then the RTC is programmed over the I2C bus, at its address `(0x68)`.

```
void setRTC() {
  // program RTC
  Wire.beginTransmission(RTCADDR);
  Wire.write(0);                  // next input at sec register

  Wire.write(decToBcd(sec));      // set seconds
  Wire.write(decToBcd(mns));      // set minutes
  Wire.write(decToBcd(hrs));      // set hours
  Wire.write(decToBcd(dow));      // set day of week
  Wire.write(decToBcd(dy));       // set date (1 to 31)
  Wire.write(decToBcd(mth));      // set month (1-12)
  Wire.write(decToBcd(yr));       // set year (0 to 99)
  Wire.endTransmission();
}
```



A simpler sketch to just read the date & time and display it is *File > Sketchbook > My_RTC*.

Later we will se if we can include the date and time into the Signal Generator code.