Crownhill reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. Neither Crownhill Associates Limited or the author shall be responsible for any claims attributable to errors omissions or other inaccuracies in the information or materials contained in this publication and in no event shall Crownhill Associates or the author be liable for direct indirect or special incidental or consequential damages arising out of the use of such information or material. Neither Crownhill or the author convey any license under any patent or other right, and make no representation that the circuits are free of patent infringement. Charts and schedules contained herein reflect representative operating parameters, and may vary depending upon a user's specific application.

All terms mentioned in this document that are known to be trademarks or service marks have been appropriately marked. Use of a term in this publication should not be regarded as affecting the validity of any trademark.

PICmicro^{tm} is a trade name of Microchip Technologies Inc. www.microchip.com

PROTON^{tm} is a trade name of Crownhill Associates Ltd. www.crownhill.co.uk

Web url's are correct at the time of publication

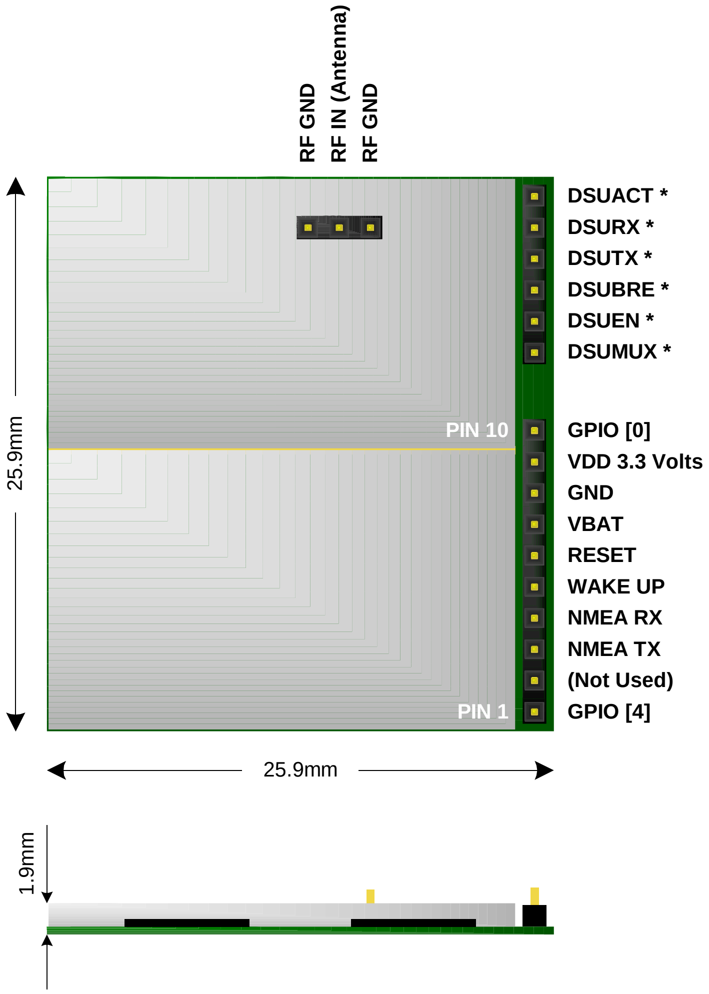
All Manufacturer Trademarks Acknowledged

ZX4120 GPS Module Interfacing.

The ZX4120 GPS module is a high sensitivity, ultra low power design squeezed into a very small package measuring only 25.9mm square and only 1.9mm high, ideal for an embedded solution.

The GPS module is based around the NEMERIX chipset, and uses 16-channels for reliable satellite tracking. Interfacing is via a regular RS232 link operating at 9600 baud (8N1), and outputs the industry standard NMEA0183 protocol.

The pinouts for the ZX4120 module is shown below.

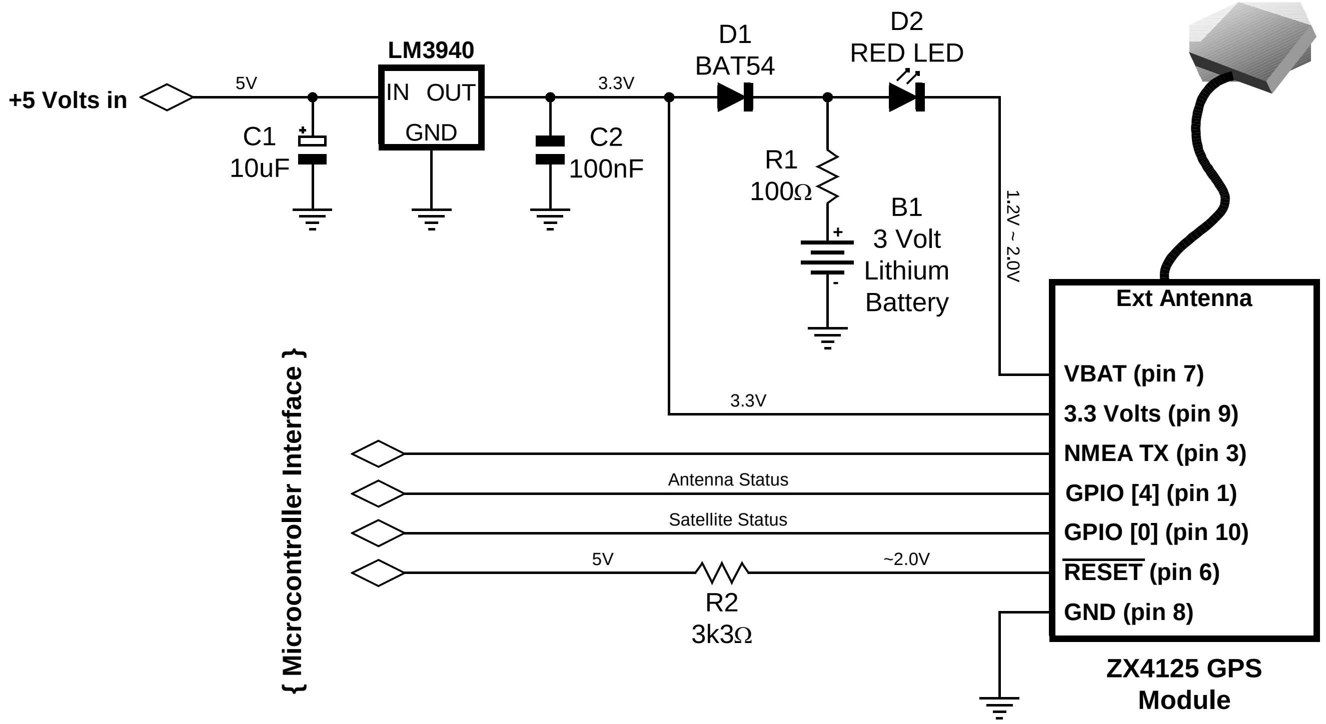


Pins DSUMUX, DSUEN, DSUBRE, DSUTX, DSURX, and DSUACT are for use with a dedicated debug unit and are not required to be connected in normal use.

ZX4120 GPS Module Interfacing.

Interfacing to a PICmicro^{tm} is simplicity itself thanks to the RS232 interface. The only thing that needs to be taken into consideration is the fact that the ZX4125 GPS module operates with a 3.3 Volt VDD, while PICmicro's normally operate at 5 Volts, but can work down to 3.3 Volts.

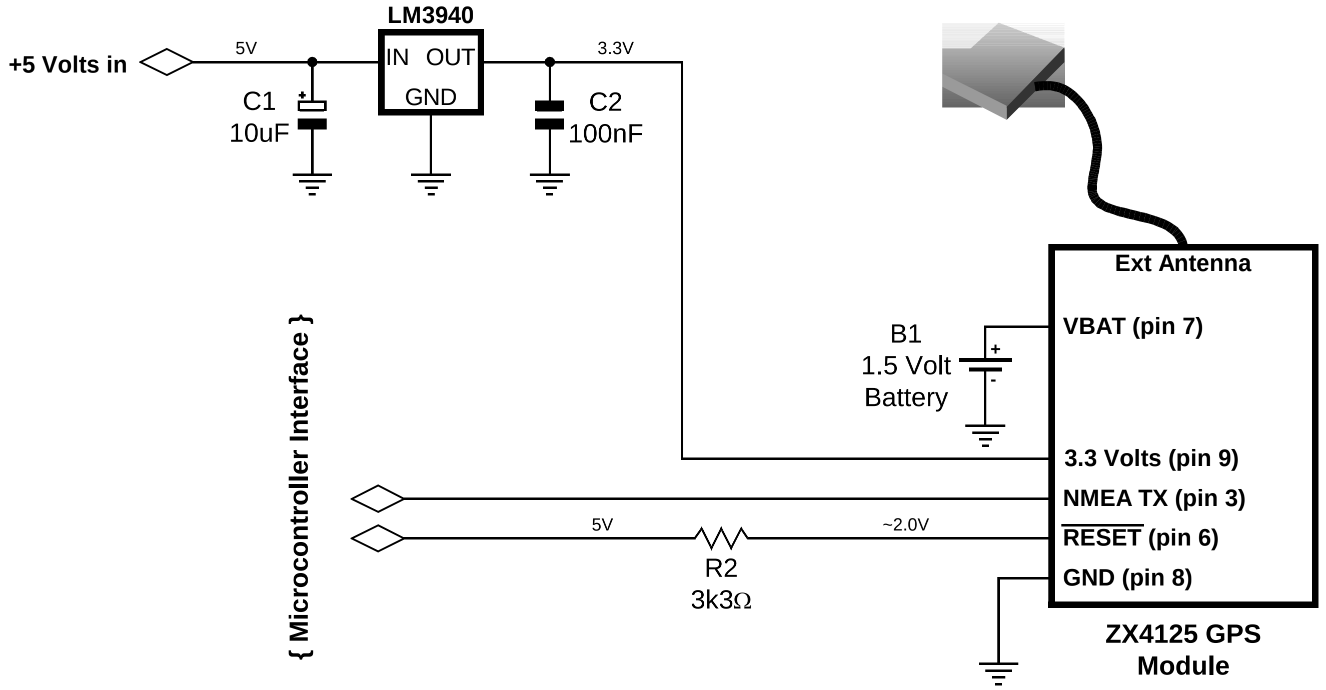
A typical reference circuit for a PICmicro^{tm} interface is shown below.



Pins GPIO [0], GPIO [4] and RESET are optional, but it is recommended to have control over the RESET line from the microcontroller in order to guarantee a clean power-up sequence.

GPIO [0] will be brought high if the antenna is shorted or open circuit.
GPIO [4] will be pulled low when the first satellite is acquired, however.

The above circuit assumes that a rechargeable battery will be used, however, a single cell 1.5 Volt non-rechargeable type can also be used with a small change of circuit, as shown below.



ZX4120 GPS Module Interfacing.

A suitable program written in PROTON+ BASIC is available for download from here: -

<Place download link here>

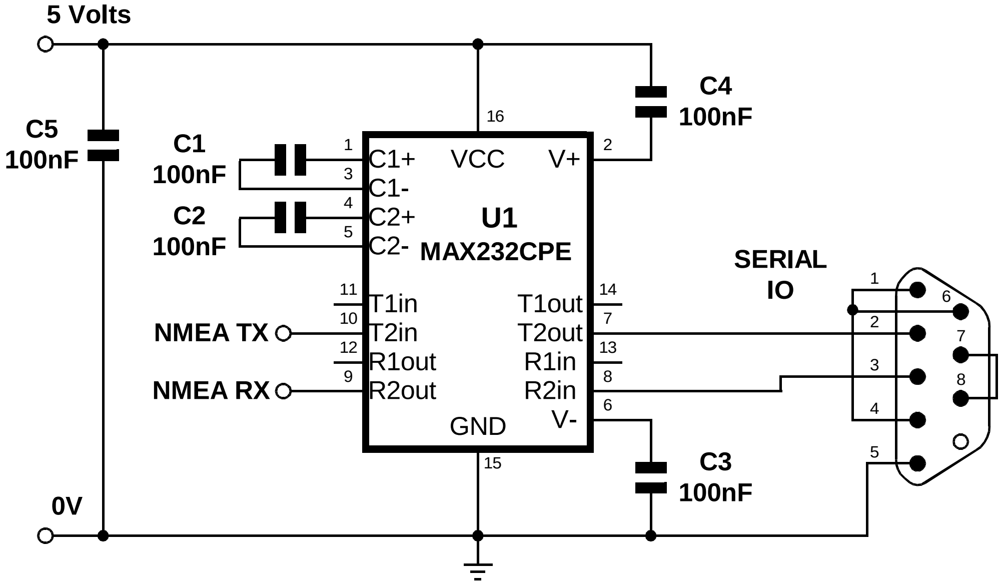
That will parse the NMEA string received from the GPS module, extract the relevant information and display it serially on any terminal. The program is named **GPS_PARSE.BAS** and it uses an include file named **GPS_NMEA_PARSE.INC** that actually contains the GPS parsing sub-routines.

In order to use the include file, place it close to the top of your main program. The listing below shows a simple demonstration program.

```
'
' NMEA decode test program
'
' Only for 16-bit core (18F) devices using version 3.1.5 of the PROTON+ compiler.
'
OPTIMISER_LEVEL = 6
Device = 18F452
XTAL = 20

HSERIAL_BAUD = 19200                ' Set baud rate
HSERIAL_RCSTA = %10010000           ' Enable serial port and continuous receive
HSERIAL_TXSTA = %00100100           ' Enable transmit and asynchronous mode
HSERIAL_CLEAR = ON                  ' Enable Error clearing on received characters

Include "GPS_NMEA_PARSE.INC"        ' Load the NMEA parse subroutines into memory

'------------------------------------------------------------------------------
MAIN_PROGRAM_LOOP:

Low GPS_RESET_PIN                   ' \
Delayms 100                         '   Reset the ZX4120 GPS module
Input GPS_RESET_PIN                 ' /

While 1 = 1                         ' Create an infinite loop
Gosub PARSE_GPS                     ' Gather the serial data and parse it
Select GPS_FIX_TYPE                 ' Check if a GPS fix has occurred
 Case 1,2,3
  Hrsout Rep "-"\10, 13
  Hrsout "DATE ", Dec2 GPS_DAY, ":",Dec2 GPS_MONTH, ":", Dec2 GPS_YEAR, 13
  Hrsout "TIME ", Dec2 GPS_HOURS, ":",Dec2 GPS_MINUTES,_
         ":", Dec2 GPS_SECONDS, " UTC", 13
  Hrsout "LAT ", Dec2 GPS_LAT_DEGREES, ",", Dec2 GPS_LAT_MINUTES, ".",_
         Dec2 GPS_LAT_SECONDS," ", GPS_LAT_BEARING, 13
  Hrsout "LON ", Dec3 GPS_LON_DEGREES, ",", Dec2 GPS_LON_MINUTES, ".",_
         Dec2 GPS_LON_SECONDS," ", GPS_LON_BEARING, 13
  Hrsout "SATELLITES IN VIEW ", Dec GPS_SATS_IN_VIEW, 13
  Hrsout "FIX TYPE ", Dec1 GPS_FIX_TYPE, 13
End Select
Delayms 700                         ' Wait before taking another GPS sample
Wend                                ' Close the loop
```

ZX4120 GPS Module Interfacing.

If the ZX4120 GPS module is being connected to a PC serial port then a serial converter will be required. The simple circuit below using a Maxim MAX232 IC may be used.



5 Volts
C5
100nF
C1
100nF
C2
100nF
C4
100nF
C3
100nF
16
1
3
4
5
2
11
10
12
9
14
7
13
8
6
15
C1+
C1-
C2+
C2-
VCC
V+
U1
MAX232CPE
T1in
T2in
R1out
R2out
T1out
T2out
R1in
R2in
GND
V-
NMEA TX
NMEA RX
SERIAL
IO
1
2
3
4
5
6
7
8
0V