

GPS Subsystem Usage

Overview

The 'OPEN GPS' and 'OPTION GPS' commands configure the PicoMite to interface with a GPS module. This integration allows the PicoMite to automatically parse NMEA data streams in the background, making location, time, and speed data available to the user program via the 'GPS()' function.

Either the 'OPEN GPS' or 'OPTION GPS' commands can be used (not both). When opened the data can then be retrieved using the GPS() function (described below).

OPEN serial-port AS GPS

```
OPEN comspec$ AS GPS [,timezone_offset] [,monitor]]
```

This command is to be included in the program and will open a serial communications port for reading from a GPS receiver (similar to opening a serial port). When opened the data can then be retrieved using the GPS() function (described below). The sentences interpreted are GPRMC, GNRMC, GPGGA and GNGGA.

'comspec\$' is the communication specification and is a string (it can be a string variable) specifying the serial port to be opened and optional parameters. It is identical to that described in Appendix A (Serial Communications) in the PicoMite User Manual. For most applications "COMn: baud" can be used where:

- 'n' is the serial port number for either COM1: or COM2:.
- 'baud' is the baud rate. This can be any number from 1200 to 921600.

The default is 9600 baud, 8 data bits, no parity and one stop bit.

The 'timezone_offset' parameter is used to convert UTC as received from the GPS to the local timezone. If omitted the timezone will default to UTC. The 'timezone_offset' can be any number between -12 and 14 allowing the time to be set correctly even for the Chatham Islands in New Zealand (UTC +12:45).

If the monitor parameter is set to 1 then all GPS input is directed to the console. This can be terminated by closing the GPS channel.

OPTION GPS

```
OPTION GPS tx_pin, rx_pin [, baud]
OPTION GPS DISABLE
```

These commands must be run at the command prompt and will trigger a soft reset of the PicoMite.

The first will configure the firmware to automatically open a serial communications port on startup for reading from a GPS. The parameters are:

- tx_pin: The GP pin number connected to the GPS module's RX pin (PicoMite Transmit).
- rx_pin: The GP pin number connected to the GPS module's TX pin (PicoMite Receive).
- baud (Optional): The baud rate for communication. Defaults to 9600 if omitted.

Note: The 'tx_pin' and 'rx_pin' must be a valid UART pair (e.g., GP0/GP1, GP4/GP5, GP8/GP9, etc.) belonging to the same UART peripheral (UART0 or UART1).

OPTION GPS DISABLE will disable the GPS functionality and release the pins for other uses.

The OPTION GPS implementation is efficient and unobtrusive to the main BASIC program.

1. Hardware Interrupts: When 'OPTION GPS' is active, the specified UART pins are reserved for the GPS. The low-level UART interrupt handlers ('on_uart_irq0' or 'on_uart_irq1' in 'Serial.c') are modified to intercept incoming data from these pins.

2. **Background Buffering:** Instead of placing the incoming characters into the standard COM port buffer, the interrupt handler diverts them into a dedicated GPS double-buffer system (`gpsbuf1` and `gpsbuf2`).
3. **Sentence Detection:** The interrupt handler monitors the data stream for newline characters. When a complete NMEA sentence is received, the current buffer is marked as "ready" (`gpsready`), and the system instantly switches to the second buffer to continue receiving data without loss.
4. **Background Parsing:** A background process (`processgps()` in `GPS.c`) periodically checks for a readybuffer. When found, it parses the NMEA sentence (supporting `\$GPGGA`, `\$GNGGA`, `\$GPRMC`, and `\$GNRMC` formats) and updates internal system variables.
5. **Data Access:** The parsed data is stored in memory and can be accessed instantly using the `GPS()` function..
6. **Timeout:** If no valid GPS data is received for 2 seconds, the system automatically marks the GPS data as invalid (`GPS(VALID)` returns 0).

Accessing GPS Data

Once configured, you can access the latest GPS data using the `GPS(item)` function as follows:

```
value = GPS(item)
```

The supported 'item' keywords are:

Item	Type	Description
LATITUDE	Float	Latitude in degrees. Positive for North, Negative for South.
LONGITUDE	Float	Longitude in degrees. Positive for East, Negative for West.
ALTITUDE	Float	Altitude in meters above sea level.
SPEED	Float	Speed over ground in knots.
TRACK	Float	Course over ground in degrees (True).
TIME	String	Current UTC time in "HH:MM:SS" format.
DATE	String	Current UTC date in "DD-MM-YYYY" format.
VALID	Integer	Returns 1 if the GPS fix is valid, 0 otherwise.
SATELLITES	Integer	Number of satellites in view/used.
FIX	Integer	Fix quality indicator (0=Invalid, 1=GPS fix, 2=DGPS fix).
DOP	Float	Dilution of Precision.
GEOID	Float	Geoidal separation in meters.

On RP2350 based PicoMites, additional astronomical calculations are available:

Item	Type	Description
SIDEREAL	Float	Local Sidereal Time in hours.
JULIAN	Float	Julian Date.

Usage Example

```
' Configure GPS on GP0 (TX) and GP1 (RX)
OPTION GPS GP0, GP1

' Wait for a valid fix
PRINT "Waiting for GPS fix..."
DO WHILE GPS (VALID) = 0
    PAUSE 1000
LOOP

' Main Loop
DO
    PRINT "Time (UTC): " + GPS (TIME)
    PRINT "Lat:  " + STR$ (GPS (LATITUDE) )
    PRINT "Long: " + STR$ (GPS (LONGITUDE) )
    PRINT "Speed:" + STR$ (GPS (SPEED) ) + " knots"
    PAUSE 5000
LOOP
```

Technical Details

- Supported NMEA Sentences:
- `GGA`: Global Positioning System Fix Data (Time, Position, Fix Type, Satellites, Altitude).
- `RMC`: Recommended Minimum Specific GNSS Data (Time, Date, Position, Speed, Track).