

OPTION GPS User Manual

Overview

The "OPTION GPS" command configures 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.

Syntax

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

- **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. Common values are 4800, 9600, 38400, etc.
- **DISABLE**: Disables the GPS functionality and releases the pins for other uses.

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).

Note: Changing this option triggers a soft reset of the PicoMite to apply the new configuration.

How It Works

The GPS implementation in PicoMite is designed to be efficient and unobtrusive to the main BASIC program.

- 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.
- 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").
- 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.
- Background Parsing**: A background process ("processgps()" in "GPS.c") periodically checks for a ready buffer. When found, it parses the NMEA sentence (supporting "\$GPGGA", "\$GNGGA", "\$GPRMC", and "\$GNRMC" formats) and updates internal system variables.
- Data Access**: The parsed data is stored in memory and can be accessed instantly using the "GPS()" function.
- 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.

Syntax

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

Supported Items

Item	Type	Description
:---	:---	:---
`LATITUDE`	Float	Latitude in degrees. Positive for North,...
`LONGITUDE`	Float	Longitude in degrees. Positive for East,...
`ALTITUDE`	Float	Altitude in meters above sea level.
`SPEED`	Float	Speed over ground in knots.

OPTION GPS User Manual

`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 oth...
`SATELLITES`	Integer	Number of satellites in view/used.
`FIX`	Integer	Fix quality indicator (0=Invalid, 1=GPS ...
`DOP`	Float	Dilution of Precision.
`GEOID`	Float	Geoidal separation in meters.

RP2350 Exclusive Items

On RP2350 based PicoMites, additional astronomical calculations are available:

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

LOCATION Command (Astronomy Context)

On RP2350 builds with astronomy support, "LOCATION" sets the observer date/time and coordinates used by astronomical commands.

Syntax

```
LOCATION "DD/MM/YYYY HH:MM:SS", latitude, longitude
LOCATION "DD/MM/YYYY HH:MM:SS", latitude, longitude, sidereal_var
```

- ****latitude****: Decimal degrees, range "-90" to "+90".
- ****longitude****: Decimal degrees, range "-180" to "+180" (East positive).
- ****sidereal_var**** (Optional): Numeric variable that receives the calculated local sidereal time (hours).

Example

```
LOCATION "25/12/2025 22:30:00", -33.86, 151.21, mydat
PRINT mydat
```

Example Usage

```
' 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
```

OPTION GPS User Manual

Technical Details

- ****Source Files****:
- "MM_Misc.c": Handles the "OPTION GPS" command parsing and configuration storage.
- "Serial.c": Manages the UART hardware interrupts and data diversion to the GPS buffers.
- "GPS.c": Contains the NMEA parser and the "GPS()" function implementation.
- ****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).