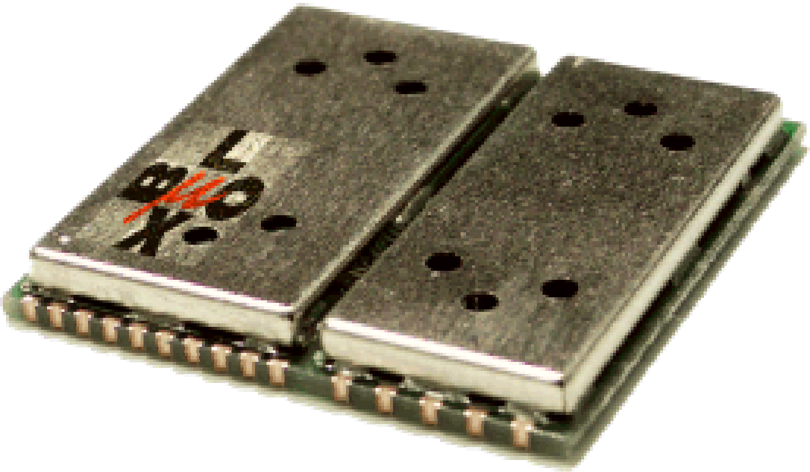
TIM

DATALOGGER

USER MANUAL



Firmware with integrated Datalogger

u-blox ag • Zuercherstrasse 68 • CH-8800 Thalwil • Switzerland • Phone +41 1 722 74 44 • Fax +41 1 722 74 47 • www.u-blox.com

Title	**TIM Datalogger Functionality**
Doc Type	USER MANUAL PRELIMINARY
Doc Id	GPS.G2-SW-02015
Author :	GvB
Date :	7. June 02
For most recent documents, please visit www.u-blox.com	

CONTENTS

1 INTRODUCTION

The firmware revision E introduces the datalogger functionality to the standard FW. u-blox offers the datalogger functionality as a standard feature in the TIM firmware.

The datalogger enables the receiver to store position, time and events in the on-board flash memory. A data logging enabled GPS receiver fulfils the specification of a standard receiver. For a description of the hardware please refer to the datasheet of the TIM GPS receiver. This document is intended to explain the concept of the logging implementation, the protocol to configure the receiver and a real world example.

The datalogger opens up a wide range of applications such as

- Vehicle tracking
- Road pricing systems
- Automatic project accounting for field personnel
- Behavioral studies of animals
- Time table analysis for public transport systems

2 FEATURES

- SW Enhancement for the u-blox TIM GPS receiver
  - Integrated datalogger in standard TIM GPS receiver firmware (Revision E and above)
  - Permanent configuration of logging parameters in Flash EPROM (using Firmware Update utility)
  - Temporary configuration in on-module RAM
  - No additional external circuitry required
- Data compression
  - Stores more than 100’000 positions (more than 5.6 Mbit Flash memory available)
- Intelligent logging algorithm triggered by GPS events based on adjustable filter parameters:
  - Elapsed time
  - Traveled distance
  - Exceeded velocity levels
- General-purpose input / output (GPIO) logging.
  - 4 independent GPIO pins available on TIM:
    GPIO[5] (pin 25), GPIO[6] (pin 24), GPIO[7] (pin 26), GPIO[10] (pin 23)
  - Pins are configurable as inputs or outputs
  - The logger observes inputs and outputs
  - Outputs may be configured high or low
- External requirements: The same as for normal operation of the TIM GPS receiver
  - One RS-232 serial interface
  - No external circuitry needed to run the data logger
  - Backup battery is not required

2.1 Required Utilities

Two u-blox utilities, which are available at the u-blox home page, are needed to configure and operate with the data logger:

Firmware Update Utility Release 1.1 (or higher) provides means to configure the datalogger and load it along with the datalogger firmware into non-volatile Flash EPROM. For details, please refer to [2].

In addition, u-Logger2.exe, provides means to retrieve logged data from the TIM GPS receiver to the PC and to make temporary configurations in RAM that is valid until power-down or reset. This utility is described in chapter 6.

2.2 New Features in the TIM Datalogger FW compared to GPS-xS1E-DL

The datalogger of the TIM GPS receivers is generally compatible to the datalogger option of the GPS-MS1E-DL and the GPS-PS1E-DL. However based on the market feedback some features have been added to the new Datalogging engine.

The datalogger storage format and the data compression algorithm is the same. The datalogger specific protocol messages are identical.

Item	**TIM, Firmware Release 2.11 UBX 1.2 or higher**	**GPS-MS1E-DL / GPS-PS1E-DL (Datalogging option)**
Datalogger functionality	Standard firmware	Optional firmware
Tools to configure data logger	Firmware Update Utility 1.1 (for permanent configuration in non-volatile Flash EPROM until a new download is made) u-Logger2.exe (for temporary configuration in volatile RAM. Data will be lost after power-down or reset)	u-logger.exe: (for configuration in battery-backed SRAM. Data will be lost if battery backup voltage fails to meet specification) Warning: Do not use u-logger2 for GPS-MS1E-DL or GPS-PS1E-DL products.
Backup battery	Not required	Required
General-Purpose I/O Ports	GPIO's 5, 6, 7 and 10	GPS-MS1E-DL: GPIO's 0…11 GPS-PS1E-DL: None
GPIO logging	On GPIO events, both GPIO logs with timestamps and position fix logs are made	Only GPIO log with timestamps are made
Storage capacity	Min. ca. 100'000 pos. fixes	GPS-MS1E-DL: ca 100'000 pos. fixes GPS-PS1E-DL: ca 20'000 pos. fixes

Table 2-1: Differences between G1 and G2 data logger

Details on configuration parameters: The datalogger configuration flags and the datalogger filter parameters are stored in a different manner.

The configuration parameters of the GPS-MS1E-DL and the GPS-PS1E-DL are stored in the Battery Backuped SRAM. In systems without a (external) backup battery, the user had to configure the unit after each power-up (or failure of the battery backup).

In the TIM GPS receiver the configuration flags and filter parameters are now stored in FLASH memory. After power on or reset the default values are copied into RAM. The configuration can be set during a firmware download. These default settings are then permanently in the code. However the parameters can be changed with configuration messages over the serial port. These changes are not permanent and will be lost after power down. The system will come up with the settings, which are stored in the FLASH memory.

3 CONFIGURATION GUIDE

The standard TIM firmware E or higher contains the datalogger as a standard feature. By default, if no configuration is made, the datalogger stays inactive to assure a fully compatible system behavior to previous firmware releases. Two programs are available to configure the data logger: The Configuration Manager (CfgMgr.exe) of the Firmware Update Utility version 1.1 [2] and u-logger2.exe.

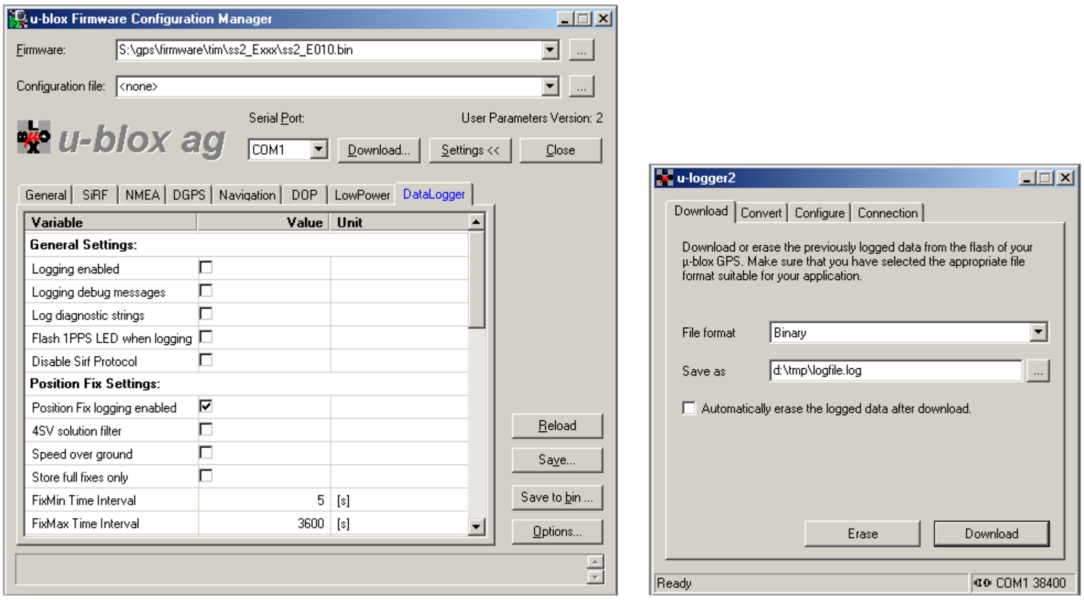


Figure 3-1: Configuration tools (left: Configuration Manager, right: u-logger2)

3.1 Configuration with Firmware Update Utility

The Firmware Update Utility (Version 1.1 or higher) [2] provides a menu to configure the datalogger and store these parameters as permanent user parameters in the Flash EPROM[1]. These parameters are retained, even if power is interrupted. After a power interruption, the firmware retrieves the user parameters and continues operation as configured.

The Configuration Manager, part of the Firmware Update Utility, provides a set of tabs to adjust user parameters. One tab is called *DataLogger Tab* and contains all relevant parameters for configuring the data logger. This tab will only be visible if a binary firmware file of release 2.11 UBX 1.2 (Exxx) or higher has been chosen.

Important Notice:

Setting configuration parameters requires downloading a new firmware image into TIM module.

3.2 Configuration with U-Logger2

The U-logger2.exe provides a menu to configure the datalogger and store these parameters as temporary parameters in the on-module volatile RAM. This configuration overrides the configuration in Flash EPROM and remains valid until power-down or system reset. U-logger2 It requires Windows 95/98, Microsoft Windows NT 4 or Microsoft Windows 2000. It requires no installation procedure to run.

Important Notice:

Configuration parameters set with u-logger2.exe and downloaded into the TIM GPS receiver will be stored in volatile static RAM and will be lost after power-down or reset.

Detailed information about u-logger2.exe is in chapter 6.2.

[1] The data logger in the first generation products GPS-MS1E and PS1E had the parameters stored in battery-backed SRAM. The configuration parameters have been retained as long backup power was available

3.3 General Data Logger: Parameters and Flags

Figure 3-2 illustrates the available configuration parameters.

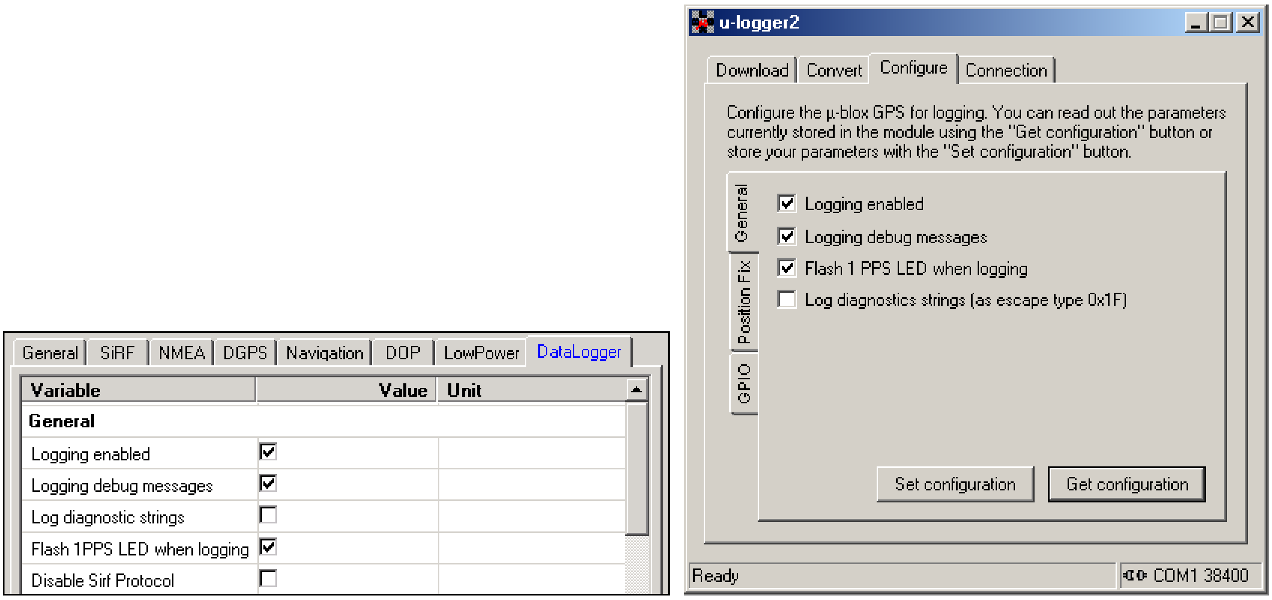


Figure 3-2: General datalogger flags (left: Configuration Manager, right: u-Logger2)

Entry	Description
Logging enabled	"Main on / off switch": Enables / disables the entire datalogger functionality. If disabled, then all other datalogger flags and settings are not effective.
Logging debug messages	If set, the GPS receiver transmits the logging debug messages with SiRF® binary message 255 (development messages). That means: additional messages are output next to other SiRF® binary messages. Example: The initialization (info on Flash EPROM, available capacity) process of the data logger, and logging events are sent out
Log diagnostic strings	If set, important events like *Reset* and version information are recorded in the flash.
Flash 1PPS LED when logging	If set, the LED indicates a logging cycle. By default it is disabled and the LED flashes at the measurement cycle.
Disable SiRF® protocol	If set, no SiRF® binary messages are transmitted (Configuration Manager only)

Table 3-1: General datalogger flags

3.4 Position Fix Logging: Parameters and Flags

Figure 3-3 illustrates the available configuration parameters.

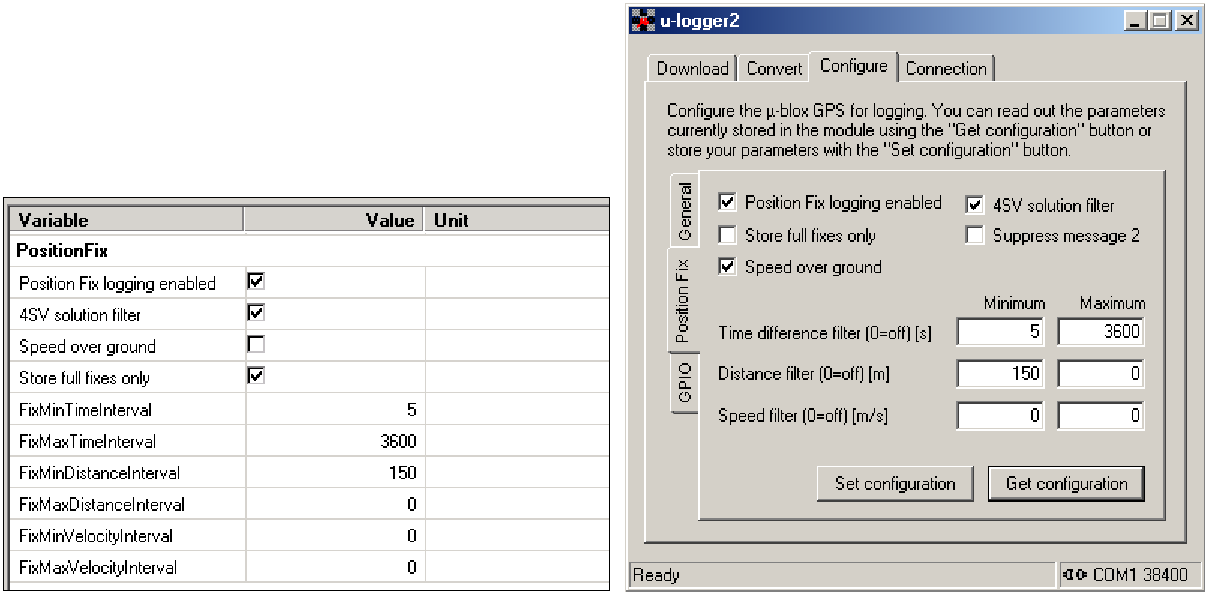


Figure 3-3: Position fix datalogger flags (left: Configuration Manager, right: u-Logger2)

Entry	Description
Position Fix logging enabled	"Main on/off switch" for position fix logging. If disabled, then all other flags and settings in this table are not effective.
4SV solution filter	If you only want to log when 3D position fixes are calculated, the 4SV solution filter has to be enabled. By default it is enabled.
Speed over ground	If disabled, then 3D velocity is used for velocity filtering criteria: FixMin/MaxVelocityInterval. If enabled, the 2D speed over ground is used.
Store full fixes only	All position fixes are stored with absolute data, requiring 18 bytes per position fix. No incremental data storage (down to 6 byte / fix) is allowed to improve utilizing data storage resources.
FixMin/MaxTimeInterval FixMin/MaxDistanceInterval FixMin/MaxVelocityInterval	The position fix filter settings (time, distance and speed filter values) are explained in chapter 5.2.1 "Position Fix Filter".
Suppress message 2	Output of SiRF® binary message 2 is suppressed. To suppress message 2 with the Configuration Manager, please refer to the "SiRF® tab" where update intervals of any supported SiRF® messages can be configured.

Table 3-2: Position fix datalogger flags

3.5 GPIO Logging: Parameters and Flags

The pins of the four supported GPIO's are illustrated in Figure 3-4.

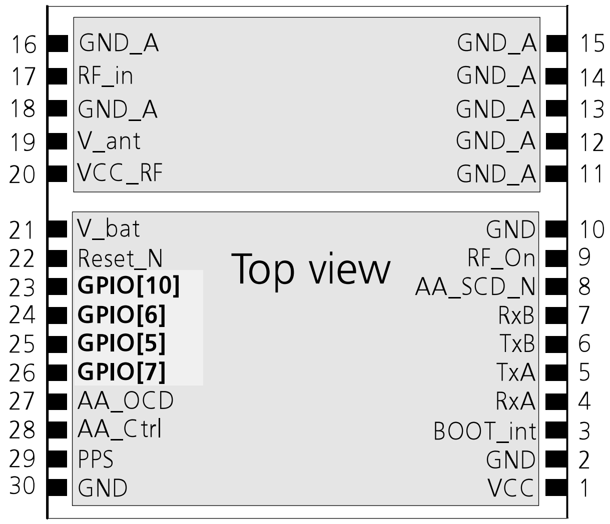


Figure 3-4: TIM module with GPIO pin assignment

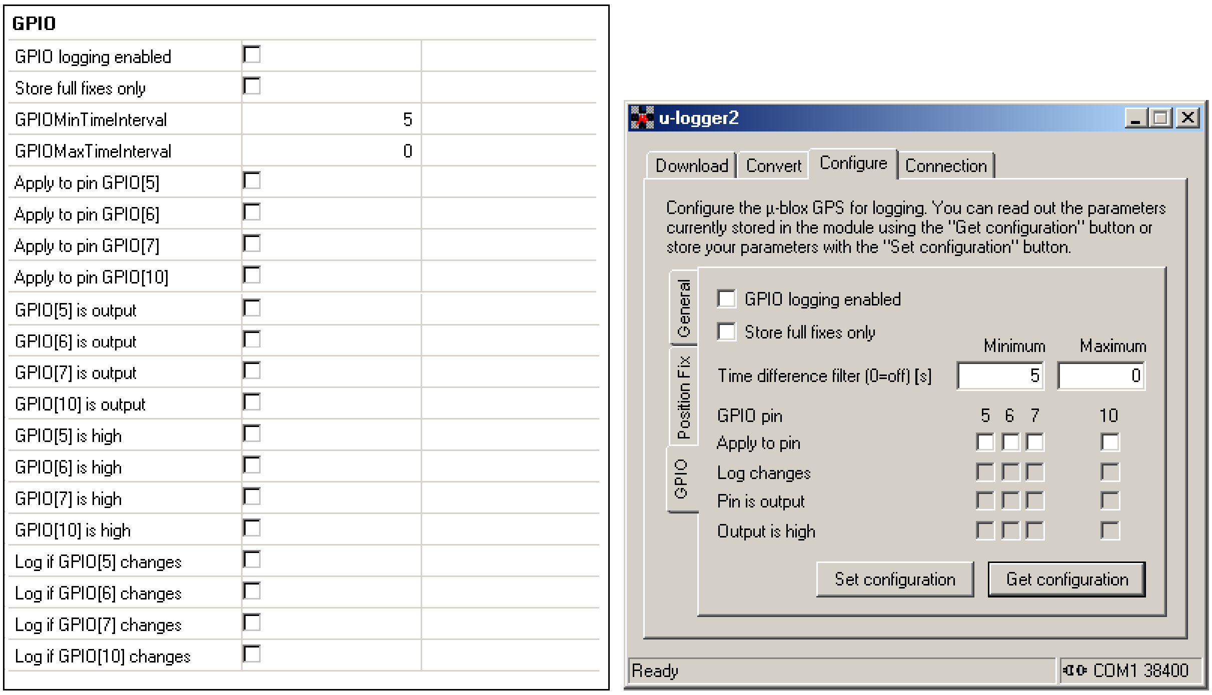


Figure 3-5: GPIO datalogger flags (left: Configuration Manager, right: u-Logger2)

Entry	Description
GPIO logging enabled	"Main on/off switch" for GPIO logging. If disabled, then all other flags and settings in this table are not effective.
Store full fixes only	If set, all position fixes are stored with absolute data, requiring 18 bytes per position fix. No incremental data storage (down to 6 byte / fix) is allowed to improve utilizing data storage resources.
GPIOMinTimeInterval	After this time and if the level at GPIO[n] has changed the datalogger does a GPIO log. If this value is zero the datalogger is able to log every second if the level at any GPIO changes.
GPIOMaxTimeInterval	After this time the datalogger does a GPIO log. If this value is zero it has no influence
Apply to pin GPIO[5,6,7,10]	Check this box if this GPIO pin is used as input or output in your system. Pins not applied are ignored for data logging.
GPIO[5,6,7,10] is output	Check this box if this GPIO is an output.
GPIO[5,6,7,10] is high	Check this box if the GPIO pin is an output und should be high. (low if disabled)
Log if GPIO[5,6,7,10] changes	Check this box if the GPIO pin is an input and the datalogger should observe this pin.

Table 3-3: GPIO datalogger flags

3.6 Protocols and Serial Ports

Once enabled the datalogger functionality in the TIM firmware works no matter which protocol is set, e.g. only NMEA output is active.

To observe debug messages and to communicate with the datalogger it is necessary to switch to SiRF® binary protocol either on port A or on port B.

3.7 TricklePower™ Mode

The datalogger works in TricklePower™ mode. Details on low power operation such as TricklePower™ mode is summarized in [5]. Following restriction applies:

The logging interval will be delayed to the TricklePower™ interval. For example, if the interval is 10 seconds and the datalogger is configured to log every 1…10 seconds, then one log is made every 10 seconds. On the other hand, if the datalogger is configured to log every 11…20 seconds, then one log is made every 20 seconds.

4 SYSTEM OVERVIEW

The software is optimized for maximum data density and maximum flexibility. A differential storage technology is used to store data in the flash memory.

The software supports two logging options that can be configured separately: Position fix logging and GPIO logging:

The information stored during position fix logging includes:

- GPS Timestamp (week number - WNO, time of week - TOW), not UTC corrected, resolution 1 second
- Position (ECEF), resolution 1 m
- Velocity, range 0…1023 km/h, resolution 1 km/h
- Number of satellites used for navigation
- DGPS used

In addition to the traditional position fix logging, the GPIO logging functionality is able store changes on the GPIO Pins. For example, a temperature sensor or an event, e.g. ignition on/off, could be logged. The information stored includes:

- GPIO signal levels, GPIO pins: GPIO[5] (pin 25), GPIO[6] (pin 24), GPIO[7] (pin 26), GPIO[10] (pin 23)
- Standard position fix logging data described above

Figure 4-1 describes the software structure of the data logger.

In case of position fix logging the GPS receivers stores data in the on board flash memory in addition to the transmission over the serial port. Basically every position fix may be stored in the Flash EPROM. But in most applications filters are used. These prevent the datalogger from storing all the positions into the Flash memory and lengthen total logging time.

Chapter 5.2 describes these filters. The user may configure the filter parameters to suit his application.

In case of GPIO logging the GPS receiver stores data on the basis of an event which recurs every second. In addition to logging it is possible to control the GPIO's. A GPIO may be used as input or output. It is also possible to set the output level to *high* (Vcc) or *low* (Gnd). Controlling the GPIOs is independent on the logging functionality.

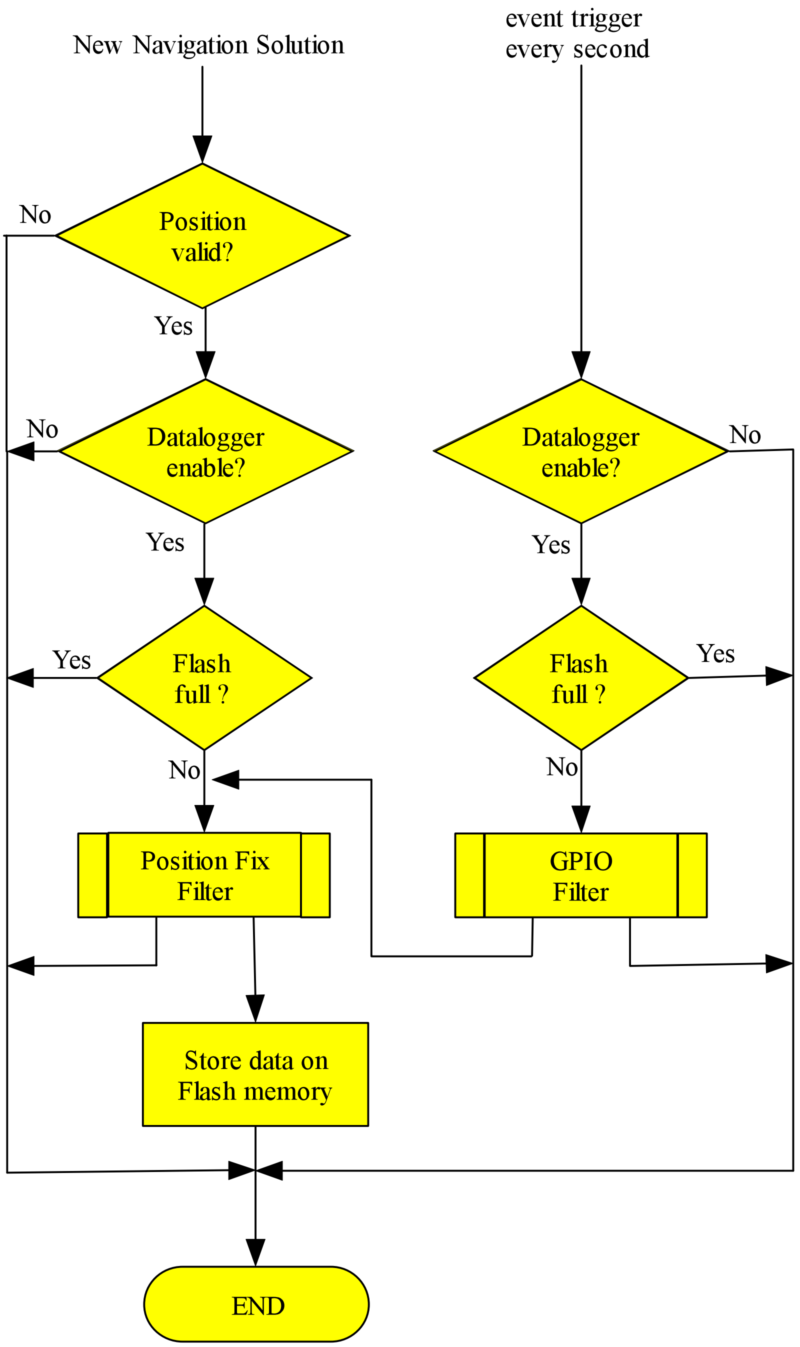


Figure 4-1: Functionality of data logger

5 Using the Data Logger

5.1 Communication with the Data Logger

The controlling of the datalogger takes place using u-blox messages in SiRF® binary protocol format via serial port (UART). Additional commands allow to adjust the logger and to download stored data.

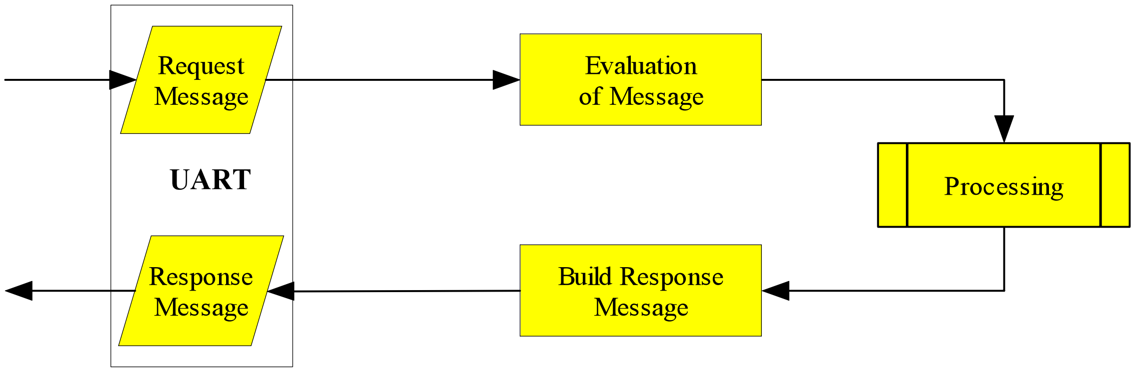


Figure 5-1: Model of data stream

Figure 5-1 shows the communication process with the data logger. The process is initiated by an incoming message. The content of the message is evaluated and processed. After processing a response is created and transmitted via UART.

The datalogger provides a new set of SiRF® binary protocol messages. The messages can be used for configuring the filter parameters. This enables user-defined position, time and velocity logging in the on board memory. Download and erasing of the flash is also supported by this protocol extension. Although the datalogger is designed as an extension to the SiRF® binary protocol, data is also stored while in NMEA mode. However, to configure the logger and download data, SiRF® binary protocol is needed.

5.2 Filter Settings

In addition to the data compression performed during the storage of a data record, the datalogger offers the possibility to further reduce the number of stored data records by configuring special filters. These filters prevent the logger from storing unnecessary data, e.g. if a vehicle is not moving. However these filters have to be set according to the requirement of the final application. The configuration is done using the additional SiRF® binary commands.

Basically one can distinguish two different types of filters: Minimum filters prevent a data record from being stored, maximum filters in contrary bypass the minimum filters, if exceeded. Therefore maximum filters can be used to make sure that data is stored, e.g. after maximum 10h.

This chapter describes the possible filter settings. An easy way to set these filters is by connecting the GPS receivers to a PC and to use the 'u-Logger2.exe' for the configuration. In an embedded environment the configuration could be set by a controller, which sends the according SiRF® binary message to the GPS receiver.

5.2.1 Position Fix Filter

This filter is active only if a new and valid position fix has been calculated. The position fix logging algorithm stores the following information:

- Timestamp of stored position, Resolution 1 second
- Velocity. Range 0 ...1023 km/h , Resolution 1 km/h
- Position. Full ECEF Position. Resolution 1 meter
- Number of SVs (<3, 3, 4 or >4 SVs).
- DGPS signal used.

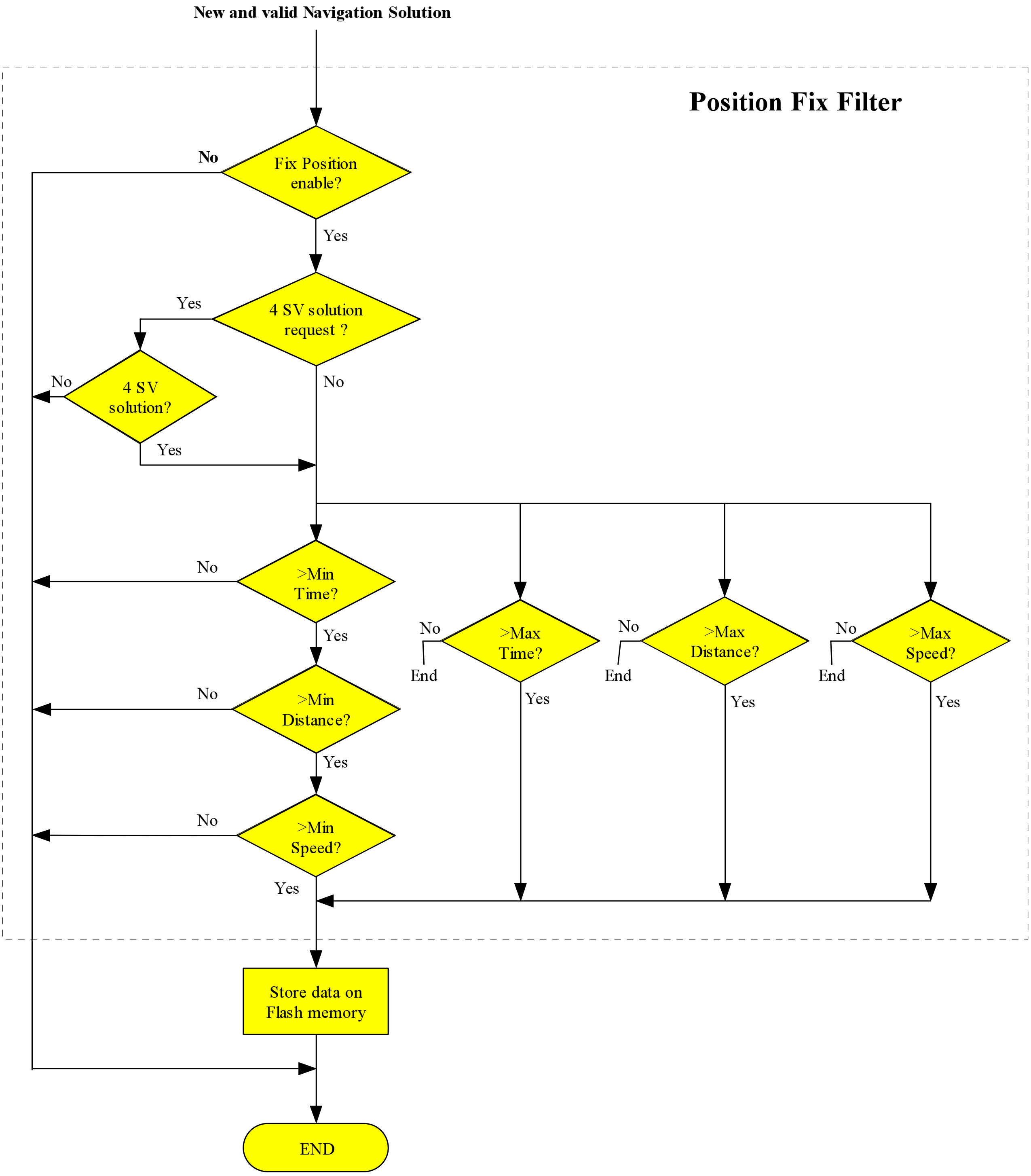


Figure 5-2: Position fix filter

5.2.1.1 Position Fix Filter Algorithm in Pseudo Code

The algorithm stores according to the following pseudo code, which is called whenever a position fix is done:

```
--Calculate the difference between now and the last storage time.
T_Diff = Current.Time - Last.FixTime

--Calculate the difference between here and the last stored position.
D_Diff = ABS(Current.Position - Last.Position)

--Get the current speed.
V = Current.Speed

--Only store if the filter checks are ok.
--The lower bounds are anded the higher bounds are ored.
IF (((T_Diff > T_Min) AND (D_Diff > D_Min) AND (V > V_Min)) OR
    (T_Diff > T_Max) OR (D_Diff > D_Max) OR (V > V_Max)) THEN

  IF ((D_Diff > 32767) OR (T_Diff > 65535)) THEN
    --Store a FIX_FULL record to the flash.
  ELSE IF (D_Diff > 511) THEN
    --Store a FIX_INCL record to the flash.
  ELSE IF (D_Diff > 15) THEN
    --Store a FIX_INCM record to the flash.
  ELSE
    --Store a FIX_INCS record to the flash.
  END IF

  --Backup storage time and position.
  Last.FixTime = Current.Time
  Last.Position = Current.Position

END IF
```

5.2.2 GPIO Filter

The GPIO logger is invoked every second. During storage, all Pin states will be saved. See chapter 5.5.3 for the definition of the storage format.

The GPIO logging algorithm stores the following information:

- Timestamp of stored position, Resolution 1 second
- Values of all GPIO pins: GPIO[5], GPIO[6], GPIO[7], GPIO[10]
- Position fix log (see chapter 5.2.1)

The filter sets a time and event mask that controls the storage.

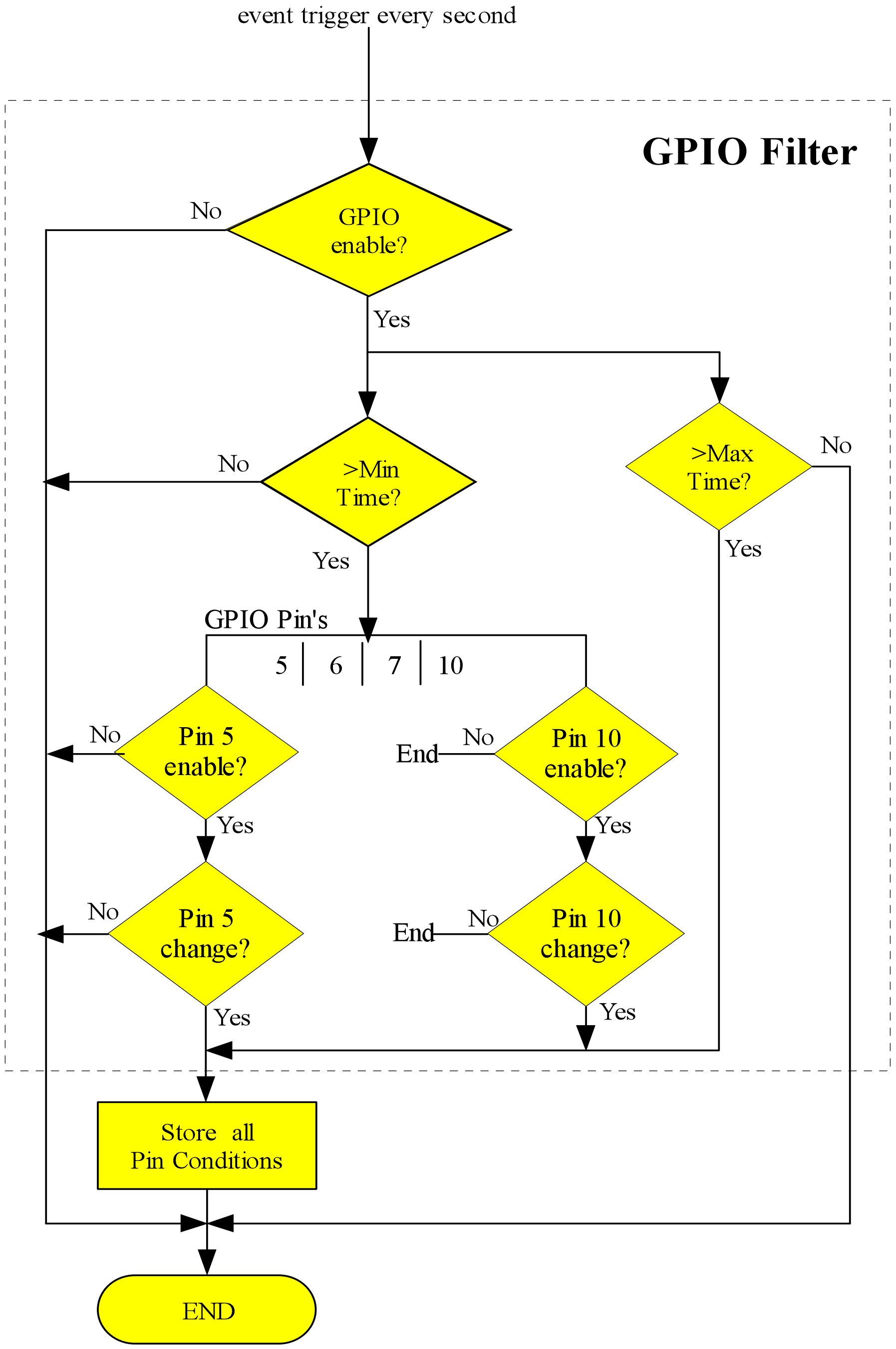


Figure 5-3: GPIO filter

5.2.2.1 GPIO Filter Algorithm in Pseudo Code

The algorithm stores according to the following pseudo code, which is called once every second:

```
-- Calculate the difference between now and the last storage time.
T_Diff = Current.Time - Last.GPIOTime

-- Only store if the filter checks are ok.
-- The lower bounds are anded the higher bounds are ored.
IF ((((Current.GPIOValue & Mask) <> (Last.GPIOValue & Mask)) AND
    (T_Diff > T_Min)) OR (T_Diff > T_Max)) THEN

  IF (T_Diff > 65535) THEN
    --Store a GPIO_FULL record to the flash.
  ELSE
  --Store a GPIO_INC record to the flash.
  END IF

  --Make position fix log
```

```
    log_position_fix();

  --Backup the storage time and GPIO values.
  Last.GPIOTime = Current.Time
  Last.GPIOValue = Current.GPIOValue

END IF
END IF
```

5.3 Default Settings of the Data Logger

The standard TIM firmware modified with the varblock_DL.ini file has the following default settings:

Validity	Parameter	Value	Protocol
General Settings	Logging enabled	Disabled	MID 0xBC
	Logging debug messages	Disabled	
	Log diagnostic strings	Disabled	
	Flash 1PPS LED when logging	Disabled	
	Disable SiRF® protocol	Disabled	
	Suppress SiRF® message 2	Disabled	
Position Fix Filter	Flags	Position Fix Logging enabled Other: disabled	MID 0xBE
	Min Time	5 s	
	Max Time	3600 s	
	Min Distance	150 m	
	Max Distance	0 m (Disabled)	
	Min Speed	0 m/s (Disabled)	
	Max Speed	0 m/s (Disabled)	
GPIO Filter	Flags	All disabled	MID 0xCA
	Min Time	5 s	
	Max Time	0 s (Disabled)	
	Mask (Configuration mask)	0x00 (None)	
	Check (Logging mask)	0x00 (Not Set)	
GPIO Settings	GPIO Logging Enabled	0x00 (Not Set)	
	Value mask	0x00 (Not Set)	

Table 5-1: Datalogger default settings

Table 5-1 describes the default settings of the data logger. The column 'Protocol' refers to these messages, which can change the settings. These messages are used to control the data logger, e.g. switch it on/off, change the settings.

The logger starts automatically during the first system start. Only the filters with above described settings will be active.

5.4 Protocol Extension

The logging protocol extension can be used with SiRF[®] binary protocol only. Please refer to the µ-blox protocol specification [2]. This document describes the payload portion of the extended SiRF[®] binary protocol, only. The following input[1]- and output messages are supported:

5.4.1 Input Messages

MID	Message	Description
0xB6	LogSectorErase (responses with LogSectorEraseEnd)	Erases all sectors or a specified sector in the flash memory
0xB8	LogRead (responses with LogData)	Initiates data download from a specified address
0xBA	LogPollSectorInfo (responses with LogSectorInfo)	Requests flash sector information
0xBB	LogPollInfo (responses with LogInfo)	Requests information about flash memory and logging space
0xBC	LogSetConfig	Sets general logging configuration
0xBD	LogPollConfig (responses with LogConfig)	Requests general logging configuration
0xBE	LogFixSetConfig	Sets the position fix logging configuration
0xBF	LogFixPollConfig (responses with LogFixConfig)	Requests the position fix logging configuration
0xC0	LogGPIOSetConfig	Sets the GPIO logging configuration
0xC1	LogGPIOPollConfig (responses with LogGPIOConfig)	Requests the GPIO logging configuration

Table 5-2: Input messages

5.4.1.1 LogSectorErase

This message causes the receiver to erase a specific flash sector. The receiver disables flash writing. After erasing the receiver returns a message of type LogSectorEraseEnd (0x7B). After erasing sectors you must reset the receiver. Send the navigation initialization message (MID = 0x80). There are two special sector numbers that erase all sectors in a row. If you send the message with 0xFF as sector number, the module will erase all used sectors, then it replies with the LogSectorEraseEnd message and performs a reset. If you send the message with 0xFE as Sector Number, the module will erase all sectors regardless of the usage, then it replies with the LogSectorEraseEnd message and performs a reset. Keep in mind that the erase command may take several seconds to complete. During this time no communication is possible.

Field	Type	Description
MID	U8	0xB6
Sector	U8	Sector number
		Payload: 2 bytes

Table 5-3: LogSectorErase message

[1] Input as seen from the receiver, i.e. from the host PC to the µ-blox receiver.

5.4.1.2 LogRead

This message requests 512 bytes of stored and compressed log data. The module returns a message of type LogData (0x79)

Field	Type	Description
MID	U8	0xB8
Address	U32	Address from which data should be returned
Payload: 5 bytes		

Table 5-4: LogRead message

5.4.1.3 LogPollSectorInfo

This message requests information on a specific sector of the flash memory. The receiver returns a message of type LogSectorInfo (0x7A).

Field	Type	Description
MID	U8	0xBA
Sector	U8	Sector number
Payload: 2 bytes		

Table 5-5: LogPollSectorInfo message

5.4.1.4 LogPollInfo

This message requests information on flash memory and logging space. The receiver returns a message of type LogInfo (0x7C).

Field	Type	Description
MID	U8	0xBB
Payload: 1 Byte		

Table 5-6: LogPollInfo message

5.4.1.5 LogSetConfig

This message sets the general logging configuration.

Field	Type	Description
MID	U8	0xBC
Flags	U16	Logging flags. See Table 5-8 for meaning
Payload: 3 bytes		

Table 5-7: LogSetConfig message

Bit #	Meaning	Parameters
Bit 0	Logging Control	0=Disabled 1=Enabled
Bit 1	Logging Debug Messages	0=Disabled 1=Enabled
Bit 2	Logging Diagnostics Strings	0=Disabled 1=Enabled
Bit 7	Flash 1PPS LED when logging	0=Disabled 1=Enabled

Table 5-8: LogSetConfig.Flags bitmap

5.4.1.6 LogPollConfig

This message requests the general logging configuration. The receiver returns a message of type LogConfig (0x7D).

Field	Type	Description
MID	U8	`0xBD`
		Payload: 1 Byte

Table 5-9: LogPollConfig message

5.4.1.7 LogFixSetConfig

This message sets the position fix logging configuration. The lower bounds (min) of the filter parameters are AND-ed, and the higher bounds are OR-ed.

Field	Type	Description
MID	U8	`0xBE`
Flags	U16	*Fix Logging Flags*. See Table 5-11 for meaning
T_min [s]	U16	*Time difference filter*. This field sets the minimum time difference with which a record maybe stored. 0=disabled
T_max [s]	U16	*Time difference filter*. This field sets the maximum time difference with which a record is stored regardless from the other parameters. 0=disabled
D_min [m]	U16	*Distance filter*. This field sets the minimum distance with which a record may be stored. 0=disabled
D_max [m]	U16	*Distance filter*. This field sets the maximum distance with which a record is stored regardless from the other parameters. 0=disabled
V_min [m/s]	U16	*Velocity filter*. This field sets the minimum speed with which a record may be stored. 0=disabled
V_max [m/s]	U16	*Velocity filter*. This field sets the maximum speed with which a record is stored regardless from the other parameters. 0=disabled
		Payload: 15 bytes

Table 5-10: LogFixSetConfig message

Bit #	Meaning	Parameters
Bit 0	Position Fix Logging Control	0=Disabled 1=Enabled
Bit 2	Output Measured Navigation on Serial Port (SiRF® Binary Message 2) while Logging	0=Output 1=Don't Output
Bit 3	Log Filter for 4SV Solution	1=Log only if 4 or more SV used 0=Log if valid navigation solution
Bit 6	Speed Format	0=3D Speed 1=2D Speed, Speed over ground
Bit 7	Store FULL records only	0=Compressed 1=Uncompressed

Table 5-11: LogFixSetConfig.Flags bitmap

5.4.1.8 LogFixPollConfig

This message requests the position fix logging configuration. The receiver returns a message of type LogFixConfig (0x7E).

Field	Type	Description
MID	U8	`0xBF`
		Payload: 1 Byte

Table 5-12: LogFixPollConfig message

5.4.1.9 LogGPIOSetConfig

This message sets the GPIO logging configuration. The lower bound (min) of the time filter is AND-ed with the gpio filter, the higher bound is OR-ed.

Field	Type	Description
MID	U8	`0xC0`
Flags	U16	*GPIO Logging Flags*. See Table 5-14 for meaning
T_min [s]	U16	*Time difference filter*. This Field sets the minimum time difference with which a record maybe stored. 0=disabled
T_max [s]	U16	*Time difference filter*. This Field sets the maximum time difference with which a record is stored regardless from the gpio filter parameters. 0=disabled
Mask	U16[3]	*Pin Mask*, Any modification applies to the here masked pins only. (1 = Change, 0 = Leave)
Direction	U16[3]	*Direction Bitmask* (1=Output, 0=Input)
Value	U16[3]	*Value Bitmask* (1 = High, 0 = Low)
Check	U16[3]	*Check Bitmask* (1=Log if Pin changes)
		Payload: 15 Bytes

Table 5-13: LogGPIOSetConfig message

Bit #	Meaning	Parameters
Bit 0	GPIO Logging Control	0=Disabled 1=Enabled
Bit 7	Store FULL records only,	0=Compressed 1=Uncompressed

Table 5-14: LogGPIOSetConfig.Flags bitmap

[3] Bitmask: the bit X represents GPIO X, only bits 5,6,7 and 10 are supported

5.4.1.10 LogGPIOPollConfig

This message requests the GPIO logging configuration. The receiver returns a message of type LogGPIOConfig (0x7F).

Field	Type	Description
MID	U8	`0xC1`
		Payload: 1 Byte

Table 5-15: LogGPIOPollConfig message

5.4.2 Output Messages

MID	Message	Description
0x79	LogData (response to LogRead)	Logged data
0x7A	LogSectorInfo (response to LogPollSectorInfo)	Sector information
0x7B	LogSectorEraseEnd (response to LogSectorErase)	Indicates the end of a sector erase
0x7C	LogInfo (response to LogPollInfo)	Contains information about flash architecture and logging space
0x7D	LogConfig (response to LogPollConfig)	Contains the general logging configuration
0x7E	LogFixConfig (response to LogFixPollConfig)	Contains the position fix logging configuration
0x7F	LogGPIOConfig (response to LogGPIOPollConfig)	Contains the GPIO logging configuration

Table 5-16: Output messages

5.4.2.1 LogData

This message is sent as a response to a LogRead message.

Field	Type	Description
MID	U8	`0x79`
Start	U32	Start address of this 512 Byte Block.
Data[256]	256 x U16	Compressed Data See chapter 'Storage Format' for a description of the compressed data structures
		Payload: 517 Bytes

Table 5-17: LogData message

5.4.2.2 LogSectorInfo

This message is sent as a response to a LogPollSectorInfo message.

Field	Type	Description
MID	U8	`0x7A`
Sector	U8	sector number
Flags	U16	(reserved)
Size	U32	Size of this sector in bytes.
Base	U32	Start address of this sector. To be used with LogRead.
Free	U32	Number of bytes available in this sector.
		Payload: 16 Bytes

Table 5-18: LogSectorInfo message

5.4.2.3 LogSectorEraseEnd

This message is sent as a response to a LogSectorErase message.

Field	Type	Description
MID	U8	`0x7B`
Sector	U8	Sector number
		Payload: 2 Bytes

Table 5-19: LogSectorEraseEnd message

5.4.2.4 LogInfo

This message is sent as a response to a LogPollInfo message.

Field	Type	Description
MID	U8	`0x7C`
S_First	U8	Index of first sector of the available logging space (zerobased)
S_Last	U8	Index of last sector of the available logging space (zerobased)
A_First	U32	First address in the logging space.
A_Last	U32	Last address in the logging space.
A_Start	U32	Start address of the used logging space.
Size	U32	Size of the used logging space.
		Payload: 19 Bytes

Table 5-20: LogInfo message

5.4.2.5 LogConfig

This message is sent as a response to a LogPollConfig message.

Field	Type	Description
MID	U8	`0x7D`
Flags	U16	See LogSetConfig message.
		Payload: 3 Bytes

Table 5-21: LogConfig message

5.4.2.6 LogFixConfig

This message is sent as a response to a LogFixPollConfig message.

Field	Type	Description
MID	U8	`0x7E`
Flags	U16	See LogFixSetConfig message.
T_min [s]	U16	See LogFixSetConfig message.
T_max [s]	U16	See LogFixSetConfig message.
D_min [m]	U16	See LogFixSetConfig message.
D_max [m]	U16	See LogFixSetConfig message.
V_min [m/s]	U16	See LogFixSetConfig message.
V_max [m/s]	U16	See LogFixSetConfig message.
		Payload: 15 Bytes

Table 5-22: LogFixConfig message

5.4.2.7 LogGPIOConfig

This message is sent as a response to a LogGPIOPollConfig message.

Field	Type	Description
MID	U8	`0x7F`
Flags	U16	See LogGPIOSetConfig message.
T_min [s]	U16	See LogGPIOSetConfig message.
T_max [s]	U16	See LogGPIOSetConfig message.
Mask	U16	See LogGPIOSetConfig message.
Direction	U16	See LogGPIOSetConfig message.
Value	U16	See LogGPIOSetConfig message.
Check	U16	See LogGPIOSetConfig message.
		Payload: 15 Bytes

Table 5-23: LogGPIOConfig message

5.4.3 Transferring Logged Data using the Extended Protocol

```
-- get information on the flash structure.
-- send the message LogPollInfo and receive the message LogInfo.

-- allocate the required memory to store the data.
Data = MEMALLOC(Size)

-- now download all the data.
Address = A_Start

WHILE (Address < A_Start + Size)

  -- now download the block from address Address.
  -- send the message LogRead and receive the message LogData.
  -- copy the received block to its position in Data.

  -- calculate the starting address of the next block to download.
  Address = Address + 512

END WHILE

-- decompress Data.
-- use the algorithm given in charter 'Decompressing a downloaded memory block'.
```

5.5 Storage Format

The logged data is stored in the flash memory in different storage records. The logging algorithms automatically choose the type of the storage record. The data compression may be switched off by setting the 'store FULL records only' flag. Separate chapters describe the logging algorithms and the decompressing.

The three most significant bits (bits 15 to 13) determine the type of the storage record. In addition to the basic types, a flexible storage record, the so-called escape type storage record, is defined for future logging applications.

3 Bits[15:13]	Type	Size [WORDS]	Description
111	NONE	1	No or unwritten data
100	FIX_FULL	9	Position Fix data, Full storage format
010	FIX_INCL	5	Position fix data, Large incremental Storage format
000	FIX_INCM	4	Position fix data, Medium incremental Storage format
110	FIX_INCS	3	Position fix data, small incremental storage format
101	GPIO_FULL	3	GPIO data, Full storage format
011	GPIO_INC	2	GPIO data, Incremental storage format
001	ESCAPE	var	Used for Future logging Applications

Table 5-24: Storage types

5.5.1 Empty Storage Record

If the first three bits are all '1', then the word is considered as unwritten data and is skipped therefore. The following table lists the layout of an empty storage record in memory.

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	1	1	1	x[4]	x	x	x	x	x	x	x	X	x	x	x	x

Table 5-25: Storage type NONE

[4] Reserved

5.5.2 Position Fix Storage Records

The datalogger has different kinds of position fix storage records. The following parameters are stored:

Field name	Size	Description	Unit
WNO	10 Bit	Week Number (in GPS notation)	Week
TOW	20 Bit	Time of Week (in GPS notation)	Seconds
DTOW	16 Bit	Difference between last and current TOW	Seconds
ECEF_X/Y/Z	32 Bit[5]	Position in ECEF X/Y/Z Coordinate	Meters
DECEF_X/Y/Z	5/10/16[6] Bit[7]	Difference between last and current ECEF X/Y/Z Coordinate	Meters
V	10 Bit	Velocity[8]	kmh
SV	2 Bit	Number of satellites. See table 4 for meaning.	
DGPS	1 Bit	Differential GPS (1 = used, 0 = not used).	

Table 5-26: Position fix logging parameters

SV[1:0]	Symbol	Description
0 0	1D	less than 3 satellites used or Dead Reckoning
0 1	2D	3 satellites used
1 0	3D	4 or more satellites used
1 1	3D+	5 or more satellites used and fix is validated

Table 5-27: SV bit description

The following tables list the layout of the position fix storage records in memory.

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	1	0	0	SV[1:0]		DGPS	V[9:0]									
Word 1	WNO[9:0]										x[9]		TOW[19:16]			
Word 2	TOW[15:0]															
Word 3	ECEF_X[31:16]															
Word 4	ECEF_X[15:0]															
Word 5	ECEF_Y[31:16]															
Word 6	ECEF_Y[15:0]															
Word 7	ECEF_Z[31:16]															
Word 8	ECEF_Z[15:0]															

Table 5-28: Storage type FIX_FULL

[5] Signed Integer
[6] size depends on storage format
[7] Signed Integer
[8] absolute speed or speed over ground, depending on the flags in the configuration.
[9] reserved

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	0	1	0	SV[1:0]		DGPS	V[9:0]									
Word 1	DTOW[15:0]															
Word 2	DECEF_X[15:0]															
Word 3	DECEF_Y[15:0]															
Word 4	DECEF_Z[15:0]															

Table 5-29: Storage type FIX_INCL

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	0	0	0	SV[1:0]		DGPS	V[9:0]									
Word 1	DTOW[15:0]															
Word 2	DECEF_Z[5:0]						DECEF_X[9:0]									
Word 3	X[10]		DECEF_Z[9:6]				DECEF_Y[9:0]									

Table 5-30: Storage type FIX_INCM

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	1	1	0	SV[1:0]		DGPS	V[9:0]									
Word 1	DTOW[15:0]															
Word 2	x[11]	DECEF_Z[4:0]					DECEF_Y[4:0]					DECEF_X[4:0]				

Table 5-31: Storage type FIX_INCS

5.5.3 GPIO Storage Records

The datalogger has different kinds of GPIO storage records. The following parameters are stored:

Field name	Size	Description	Unit
WNO	10 Bit	Week Number (in GPS notation)	Week
TOW	20 Bit	Time of Week (in GPS notation)	Seconds
DTOW	16 Bit	Difference between last and current TOW	Seconds
GPIO	12 Bit	Values of the GPIO Pins 11 to 0	

Table 5-32: GPIO logging parameters

The following tables list the layout of the GPIO storage recods in memory.

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	1	0	1	x	GPIO[11:0]											
Word 1	WNO[9:0]										x[12]		TOW[19:16]			
Word 2	TOW[15:0]															

Table 5-33: Storage type GPIO_FULL

[10] reserved
[11] reserved
[12] reserved

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	0	1	1	X	GPIO[11:0]											
Word 1	DTOW[15:0]															

Table 5-34: Storage type GPIO_INC

5.5.4 ESCAPE Type Storage Records

ESCAPE type storage records are defined for future use of the logging firmware. They have a flexible format. Its size can be determined by the second byte (SIZE field). For example diagnostics strings may be written to the flash memory as ESCAPE_TYPE 0x1F with a string as the payload.

The following table lists the layout of the escape storage records in the memory.

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word 0	0	0	1	ESCAPE_TYPE					SIZE (n Words)							
Word 1	Payload															
Word ..	Payload															
Word ..	Payload															
Word n	Payload															

Table 5-35: Storage type ESCAPE

5.6 Decompressing a Downloaded Memory Block

The algorithm decompresses a previously downloaded memory block.

```
-- Download the data from the GPS receiver flash.
-- Use the logging protocol extension.

-- Get the first storage record into Data.

-- Decompress all storage records while we have Data.

WHILE (Data)

  -- Now decode the storage record
  -- Get the Type bits of the storage record.

  IF (Type = EMPTY) THEN
    -- No data , just skip this word.
  ELSE IF (Type = FIX_FULL) THEN
    -- Save WNO and DTOW fields as the position fix logging time stamp
    -- Save the position , speed and the other mode flags
  ELSE IF ((Type = FIX_INCL) OR (Type = FIX_INCM) OR (Type = FIX_INCS)) THEN
    -- Add the DTOW field to the last position fix logging time stamp.
    -- Add the DECEF to the last position.
    -- Save the speed and the other mode flags.
  ELSE IF (Type = GPIO_FULL) THEN
    -- Save WNO and DTOW fields as the GPIO logging time stamp.
    -- Save the GPIO values.
  ELSE IF (Type = GPIO_INC) THEN
    -- Add the DTOW field to the last GPIO logging time stamp.
    -- Save the gpio values.
  ELSE IF (Type = ESCAPE) THEN
    -- Handle the additional ESCAPE type storage records.
  END IF

  -- The size of each storage record can be determined from the type
  -- and if it is an escape  type from the additional size field.

  -- Get the next storage record into Data.

END WHILE
```

6 TRANSFERRING LOGGED DATA USING U-LOGGER2.EXE

The µ-logger is a simple program to demonstrate and evaluate the logging capabilities of the µ-blox GPS logging firmware and the protocol extension. It allows to configure the module and to download or erase the logged data. The logged data may be stored in various formats that can be post processed by using third party programs.

The program runs on IBM compatible PCs running Microsoft Windows 95/98, Microsoft Windows NT 4 or Windows 2000. It needs an unused serial port where the µ-blox GPS receiver is connected. The status bar shows the actual connection and its current status. It also indicates the step and progress of the current operation. The user may abort any operation by pressing the Cancel button.

Important notice:

All programs settings are stored in the File u-blox.ini under the section µ-logger in your windows directory.

6.1 Communication Setup

For changing settings and downloading logged data, a communication between the GPS receiver and host PC with µ-logger has to be created. The window 'Connection' is used to set up a connection.

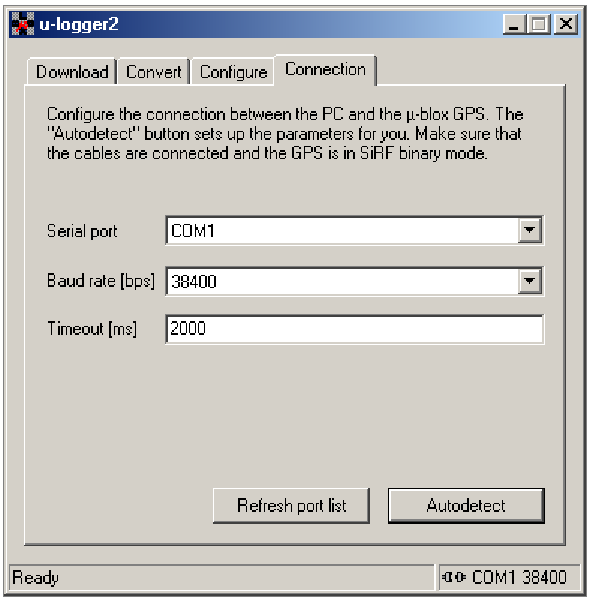


Figure 6-1: Setup communication window

Figure 6-1 shows the possibilities of settings for the serial interface. The communications port can be selected from the 'Serial Port' pull-down menu. If a port is not in the port list, another program is still connected with your GPS. Close the connection and then press the 'Refresh port list' button. The port should now appear in the port list.

The appropriate baud rate may be selected by the 'Baud rate [bps]' pull down menu. The default baud rate is 19200.

The µ-logger expects a response from the GPS receiver within the timeout value. The default is 2000 ms, it should be suitable for most applications.

Important notice:

The Auto detect button checks all serial ports and baud rates for a connected GPS. If this does not detect the GPS, make sure that it uses the SiRF® binary mode protocol and, that the cables are properly connected.

6.2 Configure the GPS Logging Parameters

For an optimised storage, several different parameters may be adjusted. Therefore the window with the different sub pages serves for this purpose.

For details on configuration parameters, please refer to section 3.2.

Important notice:

You must set or get the configuration for each sub tab separately.

The *Set configuration* button stores the parameters selected in the dialog box to the module. The *Get configuration* button reads out the configuration parameters from the module and fills them into the dialog box.

6.3 Download or Erase the Logged Data

During GPS processing, data is stored in the flash memory of the GPS module. This Window allows you to download or erase the logged data. You can choose the file formats in which the data is stored on your PC.

The datalogger works only if the flash memory has free space. Thus it has to be possible to erase the flash memory on the GPS module. Erasing takes place after download of logged data or separately.

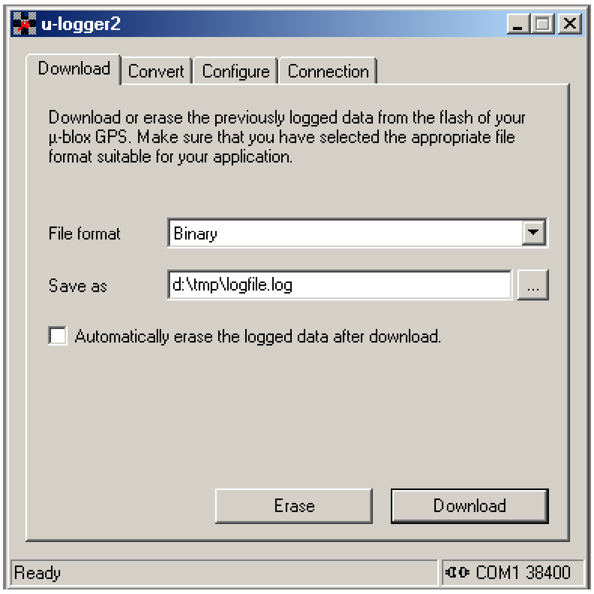


Figure 6-2: Download / erase data tab

Figure 6-2 shows the download and erase window. The following settings are possible:

- Selection of appropriate file format. Please note, that the NMEA formats contain only information of the Position Fix storage records.
- Selection of path and filename, where you want to store the File. You can press the button next to the edit box to select the path and filename.

- If you want to automatically erase the logged data after download tick the check box.
- Click the Download button to download and store the data on the disk.
- **Note:** The Erase button deletes the flash area reserved for data logging.

The following table describes the possible file formats in which the data can be stored on you PC.

Type	Description
Binary	Compressed data as in the flash, saved binary, big-endian byte order (Do not confuse with SiRF® binary protocol format)
NMEA	Decompressed data, saved as NMEA GLL, RMC, GGA and VTG messages
NMEA GLL	Decompressed data, saved as NMEA GLL messages
NMEA RMC	Decompressed data, saved as NMEA RMC messages
NMEA GGA	Decompressed data, saved as NMEA GGA messages
NMEA VTG	Decompressed data, saved as NMEA VTG messages
Text	Decompressed data, saved as ASCII text, All data
Text FIX	Decompressed data, saved as tabular ASCII text, Fix data only
Text GPIO	Decompressed data, saved as tabular ASCII text, GPIO data only

Table 6-1: Download file formats

6.4 Conversion of logged data

The logged data is transmitted within SiRF® binary message only. The logged data is stored into a file on PC with the defined format. The μ-logger offers the possibility to convert binary download files into all other described download formats.

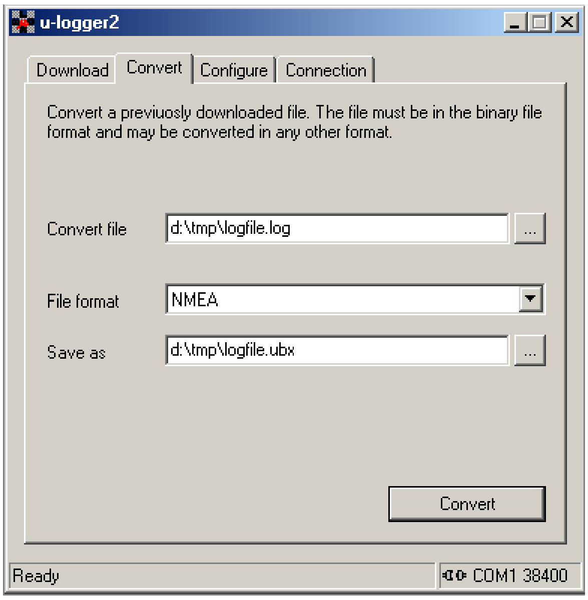


Figure 6-3: Convert tab

This picture shows the Convert Window. The file to be converted has to be filled into the field 'Convert File'. The 'File Format' describes the format of the resulting file. The field 'Save as' describes the destination file to hold the converted data.

Important notice:

This enables you to convert the downloaded data of a binary file into the different formats. The Datalogger does not need to be connected while converting.

6.5 Application Examples

This chapter helps to understand the time difference, distance and speed filters with different examples.

The lower bounds (min) of the filter parameters are AND-ed. the higher bounds are OR-ed. All parameters are 16 bits. The upper boundary of velocity is 1023 m/s. If a value is zero, then it will be ignored. If all values are zero, then no position fix logging takes place.

	Minimum (lower boundary)	**Maximum (upper boundary)**
Time difference filter	T_min (seconds)	T_max (seconds)
	AND	**OR**
Distance filter	D_min (meters)	D_max (meters)
	AND	**OR**
Speed filter	V_min (m/s)	V_max (m/s)
Combination of both	**OR**	
	Logging event	

Table 6-2: Position fix parameters

	T_min [s]	**T_max [s]**	**D_min [m]**	**D_max [m]**	**V_min [m/s]**	**V_max [m/s]**
Log every 10 seconds	0	10	0	0	0	0
Log every 100 meters	0	0	0	100	0	0
Log if velocity ≥40 m/s	0	0	0	0	0	40

Table 6-3: parameter settings focusing on single parameters

6.5.1 Hiker

A hiker wants to track his traveled path with a reasonable resolution. Log every 20 seconds if the hiker is moving. Log every 15 minutes if the hiker is in not moving.

	T_min [s]	**T_max [s]**	**D_min [m]**	**D_max [m]**	**V_min [m/s]**	**V_max [m/s]**
Hiker	20	900	20	0	0	0

Table 6-4: Hiker: recommended position fix settings

6.5.2 Tracking Vehicles

Possible requirement:

Log every 50 meters if the vehicle moves with velocity less than 25 km/h (7 m/s).
Log every two seconds if the vehicle moves between 25 km/h (7 m/s) and 50 km/h (14 m/s).
Log every second if the vehicle drives with more than 50 km/h (14 m/s).

	T_min [s]	**T_max [s]**	**D_min [m]**	**D_max [m]**	**V_min [m/s]**	**V_max [m/s]**
Tracking vehicles	2	0	0	50	7	14

Table 6-5: Tracking vehicles: recommended position fix settings

6.5.3 Tracking Rental Cars

Car rental firms may be interested to identify customers violating traffic regulations (e.g. 140 km/h or 90 mp/h exceeded where most countries rule maximum 120 Km/h or 75 mp/h) and enforcing insurance terms like forbidden entry into unauthorized countries. One position fix per hour is sufficient to identify border crossings. Exceeding speed boundary results to more frequent logging at interval of 15 seconds.

	T_min [s]	T_max [s]	D_min [m]	D_max [m]	V_min [m/s]	V_max [m/s]
Tracking rental cars	15	3600	0	0	40	0

Table 6-6: Tracking rental cars: recommended position fix settings

6.5.4 Mark Bus Stations

Tracking of a bus shall include information of the bus stops where stopped and how long. The goal is to have only position fix logs when the GPS receiver does not move for a certain time.

If you want to implement such a solution it is recommended to combine the GPIO logging and the position fix logging. One GPIO pin is going to be used as a stop indicator from the vehicle. The datalogger makes and stores a position fix when the level changes at the input pin.

Alternatively, consider logging at fixed time intervals, e.g. once every 10 seconds, storage capacity is available for more than 5 consecutive days, assuming 50'000 logs fit inside!

	T_min [s]	T_max [s]	D_min [m]	D_max [m]	V_min [m/s]	V_max [m/s]
Tracking rental cars	0	10	0	0	0	0

Table 6-7: Bus stations: recommended position fix settings

6.6 Fix Logging Performance Example

Assume that the Fix logging parameters are

```
T_min = 5[s]
T_max = 3600[s]
D_Delta = 150[m]
```

The logging time depends on the memory available for logging and on how the receiver is moved. The u-blox GPS receivers have 1`024 kBytes (8 MBits) of flash memory of which 704 kbytes may be used for logging. The following equation calculates how long data can be logged.

$$\text{logging time} = \text{time between storage} \cdot \frac{\text{free flash memory}}{\text{size per storage record}}$$

The worst case is that we would have to store a logging record in FIX_FULL format every second. The logging time will be around 11 hours. If we would store in FIX_INCS storage format the logging time is going to be around 33 hours.

Let's assume that we are constantly traveling with 50 km/h (14 m/s). The time between storage will be 11 seconds. Since we moved about 150 meters the data is most probably stored in the FIX_INCM storage record format. The calculated logging time is more than 8 days.

6.6.1 Real Example

This example shows a short ride with a car. No differential GPS was connected. The configuration used in this example:

Log every 50 meters if the vehicle moves with velocity less than 25 km/h (7 m/s).
Log every two seconds if the vehicle moves between 25 km/h (7 m/s) and 50 km/h (14 m/s).
Log every second if the vehicle drives with more than 50 km/h (14 m/s).

	T_min [s]	T_max [s]	D_min [m]	D_max [m]	V_min [m/s]	V_max [m/s]
Tracking rental cars	2	0	0	50	7	14

Table 6-8: Settings for real example

The following lines show an extract of a hexadecimal dump of the file saved in binary format.

```
00000000    9800 2445 F0EA 0041 6806 0009 D9E6 0047         ..$E...Ah......G
00000010    29BF 4800 05A5 FDA2 00EF FF15 1000 001E         ).H.............
00000020    A9F4 131C 1000 010E C7F0 3CAA 1819 0035         ..........<....5
```

The following lines show an extract of the data saved as a tabular text:

FIX_Type	Fix	DGPS	WNO	TOW	DTOW	Time	Date	Decef_X	Decef_Y	Decef_Z	Ecef_X	Ecef_Y	Ecef_Z	Speed	Longitude	Latitude	Altitude
FIX_FULL	3D+	No	145	389354	0	12:09:01	06/06/2002	0	0	0	4286470	645606	4663743	0	8.565215	47.28526	555
FIX_INCL	2D	No	145	390799	1445	12:33:06	06/06/2002	-606	239	-235	4285864	645845	4663508	0	8.569532	47.28755	0
FIX_INCM	3D	No	145	390829	30	12:33:36	06/06/2002	500	-228	298	4286364	645617	4663806	0	8.565568	47.28633	531
FIX_INCM	3D	No	145	391099	270	12:38:06	06/06/2002	-16	170	-15	4286348	645787	4663791	0	8.567821	47.28617	526
FIX_INCM	3D+	No	145	391152	53	12:38:59	06/06/2002	14	32	-12	4286362	645819	4663779	25	8.568212	47.28598	530
FIX_INCM	3D+	No	145	391154	2	12:39:01	06/06/2002	2	17	-6	4286364	645836	4663773	35	8.56843	47.28591	529
FIX_INCM	3D+	No	145	391156	2	12:39:03	06/06/2002	4	21	-7	4286368	645857	4663766	42	8.568697	47.28582	528
FIX_INCM	3D+	No	145	391158	2	12:39:05	06/06/2002	4	23	-8	4286372	645880	4663758	44	8.568989	47.28572	527
FIX_INCM	3D+	No	145	391160	2	12:39:07	06/06/2002	2	23	-8	4286374	645903	4663750	44	8.569286	47.28564	525
FIX_INCM	3D+	No	145	391162	2	12:39:09	06/06/2002	2	24	-6	4286376	645927	4663744	44	8.569596	47.28557	525
FIX_INCM	3D+	No	145	391164	2	12:39:11	06/06/2002	-1	24	-5	4286375	645951	4663739	43	8.569911	47.28552	523
FIX_INCM	3D+	No	145	391166	2	12:39:13	06/06/2002	-2	24	-4	4286373	645975	4663735	43	8.570229	47.28548	521
FIX_INCM	3D+	No	145	391168	2	12:39:15	06/06/2002	-1	23	-4	4286372	645998	4663731	42	8.570532	47.28544	520
FIX_INCM	3D+	No	145	391170	2	12:39:17	06/06/2002	-2	22	-4	4286370	646020	4663727	37	8.570823	47.28541	518
FIX_INCM	3D+	No	145	391172	2	12:39:19	06/06/2002	-1	16	-4	4286369	646036	4663723	26	8.571034	47.28538	516
FIX_INCM	3D+	No	145	391178	6	12:39:25	06/06/2002	-4	40	-7	4286365	646076	4663716	26	8.571565	47.28532	512
FIX_INCM	3D+	No	145	391191	13	12:39:38	06/06/2002	-41	13	32	4286324	646089	4663748	20	8.571816	47.28577	509
FIX_INCM	3D+	No	145	391201	10	12:39:48	06/06/2002	-36	-12	38	4286288	646077	4663786	18	8.57173	47.28625	512
FIX_INCM	3D+	No	145	391211	10	12:39:58	06/06/2002	-36	-5	36	4286252	646072	4663822	17	8.571735	47.28671	513
FIX_INCM	3D+	No	145	391222	11	12:40:09	06/06/2002	-37	-12	38	4286215	646060	4663860	19	8.571651	47.2872	515
FIX_INCM	3D+	No	145	391238	16	12:40:25	06/06/2002	-38	-14	36	4286177	646046	4663896	13	8.571543	47.28768	515
FIX_INCM	3D+	No	145	391249	11	12:40:36	06/06/2002	-36	-18	31	4286141	646028	4663927	18	8.571379	47.28812	512
FIX_INCM	3D+	No	145	391260	11	12:40:47	06/06/2002	-36	-20	34	4286105	646008	4663961	16	8.571188	47.28858	511
FIX_INCM	3D+	No	145	391271	11	12:40:58	06/06/2002	-36	-21	34	4286069	645987	4663995	18	8.570985	47.28904	509

Table 6-9: Logged data as tabular text

When the car had a velocity of 40 km/h the datalogger did every two seconds a position fix. When the car was moving with less than 25 km/h the datalogger did a position fix every 50 meters.

Figure 6-4: Route with 40 and 20 Km/h sections

The datalogger did position fix logs every two seconds when the car was moving with 40 km/h.
When the datalogger was faster than 50 km/h it was logging every second.

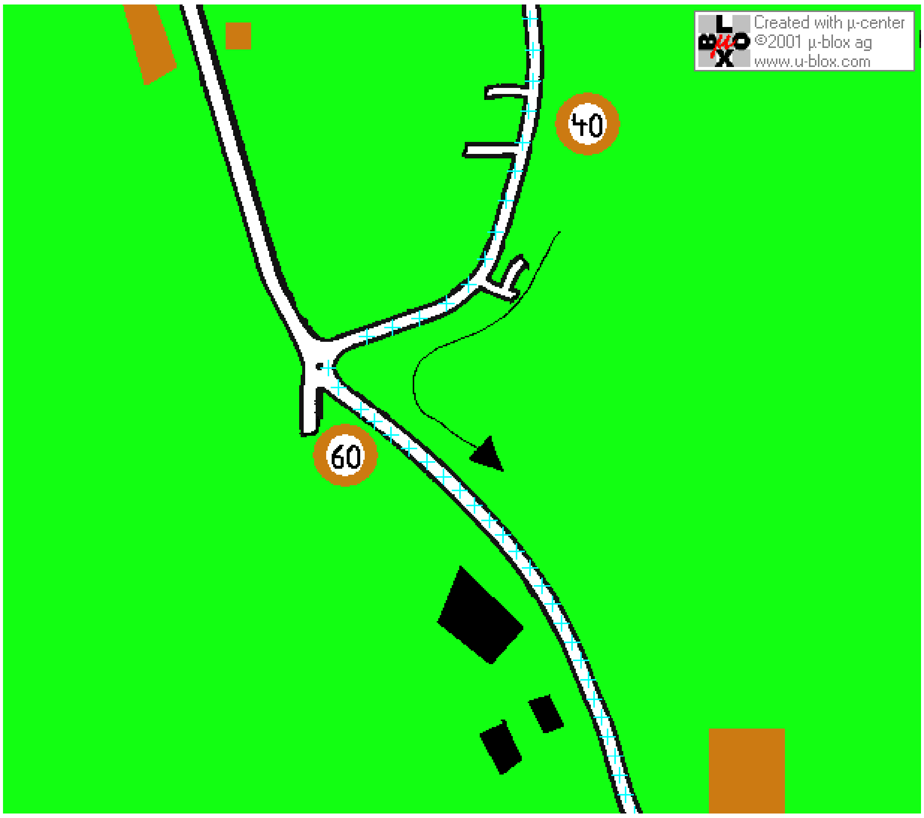


Figure 6-5: Route with 40 and 60 Km/h sections

A Related Documents

[1] TIM GPS Receiver Macro Component – Data Sheet, GPS.G2-MS2-01001

[2] TIM Firmware Update Utility – User's Manual, GPS.G2-SW-02004

[3] TIM GPS Receiver Protocol Specification– Application Note, GPS.G2-X-01003

[4] The GPS Dictionary, GPS-X-00001

[5] TIM Low Power Modes - Application Note, GPS.G2-X-02003

All these documents are available on our homepage (http://www.u-blox.com).

B Glossary

Please refer to the GPS Dictionary [4].

C Contact

For further info, please contact us:

Headquarters	Subsidiaries	
u-blox ag Zuercherstrasse 68 CH-8800 Thalwil Switzerland Phone: +41 1 722 74 44 Fax: +41 1 722 74 47 E-mail: info@u-blox.com www.u-blox.com **Tech. Support:** Phone: +41 1 722 74 74 support@u-blox.com	**u-blox Deutschland GmbH** Berliner Ring 89 D-64625 Bensheim Germany Phone: +49 (0) 6251 17566-0 Fax: +49 (0) 6251 17566-11 E-mail: info_de@u-blox.de www.u-blox.de **Tech. Support:** Phone: +41 1 722 74 74 support_de@u-blox.de	**u-blox Asia Pacific Ltd.** 22/F., City Landmark I 68 Chung On Street Tsuen Wan, Hong Kong Phone: +852-2941-8877 Fax: +852-2615-2285 E-mail: info_ap@u-blox.com www.u-blox.com **Tech. Support:** Phone: +41 1 722 74 74 support_ap@u-blox.com
	u-blox Europe Ltd. Barham Court Maidstone, Kent ME18 5BZ United Kingdom Phone: +44 1622 618628 Fax: +44 1622 618629 E-mail: info_uk@u-blox.co.uk www.u-blox.co.uk **Tech. Support:** Phone: +44 1622 618628 support_uk@u-blox.co.uk	**u-blox America, Inc.** 13800 Coppermine Road Herndon, VA 20171 USA Phone: +1 (703) 234 5290 Fax: +1 (703) 234 5770 E-mail: info_us@u-blox.com www.u-blox.com **Tech. Support:** Phone: +1 (703) 234 5290 support_us@u-blox.com