Optical Flow Sensor with IMU for Robot Navigation & Safety

Description

Roboteq's FLW100 is a high-resolution sensor especially designed for accurate contactless X-Y motion sensing over a surface. The FLW100 is intended as a navigation sensor for wheeled mobile robot.

The sensor works similarly to an optical mouse, but with higher resolution, accuracy and at greater distance from the reference surface. The sensor uses an embedded infrared camera that is pointed to the floor and measures the displacement distance and speed along the X and Y axis by comparing images at each frame. Distance is measured with 0.1mm resolution with excellent accuracy.

The sensor has a built-in infrared LED and laser illuminator and will work on practically all types of surfaces at speeds up to 1.1m/s.

In addition, the FLW100 incorporates a 9-degree of freedom sensor (gyroscope, accelerometer and magnetometer) with a fusion algorithm to turn it into an Inertial Measurement Unit (IMU) or Attitude Heading Refence System (AHRS).

The FLW100 has an 8-pin M12 industrial connector for its power supply, communication and I/O connections. The communication interface can be configured as CANbus, RS232 or serial TTL. Four I/O pins can be configured as Dual Quadrature Encoder outputs, as PWM output, or as user I/O. The FLW can be connected to a PC via its USB port for configuration and monitoring.

The sensor embeds a powerful scripting language that can be used, together with its four I/O lines, to implement safety features, such as the SS1 Safety Stop in Mobile Robots applications

Application

- Automatic Guided Vehicles
- Mobile Robots
- Warehouse Automation Robots
- Automatic Cleaning Robots

Features

- Contactless sensing of X-Y travelled distance and speed
- 0.1mm measurement resolution on each axis
- 40mm to 100mm distance from floor
- Built-in infrared LED and laser illuminator
- Tracking on glossy surfaces (metal, tiles) by LASER and rough surface (cloth, carpets) by LED with a lens
- Tracking speed is up to 1.1m/s
- Wide range, 5 to 30V power supply input
- CANbus interface with four Protocols support
  - CANopen
  - RoboCAN
  - MiniCAN
  - RawCAN
- RS232 Interface
- Serial TTL Interface
- PWM Output
- 4 User Digital Inputs/Output
- X-Y Quadrature Encoder emulation mode
- Built-in IMU/AHRS using 9DOF Accelerometer, Gyroscope, Magnetometer and fusion Algorithm
- Built-in programming language for implementing custom functionality
- USB Interface for configuration, monitoring, scripting and firmware updating
- 10-30V power supply
- Water-resistant IP65 enclosure
- Compact 105x30x34 mm design

Orderable Product

Reference	Description
FLW100	X-Y Flow sensor with built-in IMU

Connectors and LEDs identification

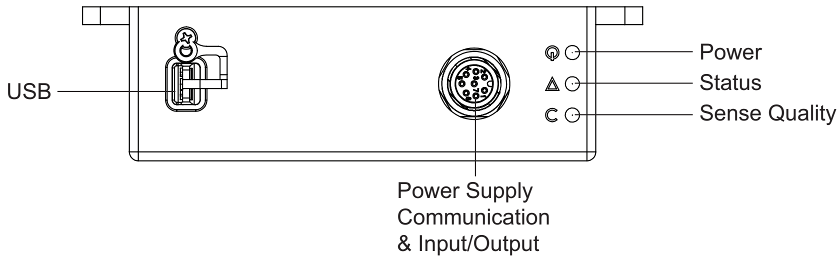


Figure xx : Connectors and LED identification

Connections

All connections to and from the sensor are done using an industrial, 8-pin M12 waterproof male connector. The eight pins consist of two Power pins, two Shared Communication pins, and four multipurpose digital Input/Output pins. The operating mode of each set of pin is configured using the sensor's PC utility.

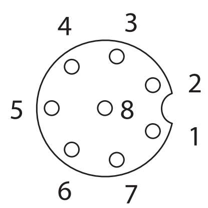


Figure xx: Connector pins identification

Table XX: Pin Assignment

Function / Pin	RS232	CAN	Q ENC	MultiPWM	Serial TTL	Digital I/O	Power
1	-	-	-	-	-	-	Vcc In
2	-	-	ENCA Y	DOUT4 / DIN4	DOUT4 / DIN4	DOUT4 / DIN4	-
3	-	-	ENCB Y	DOUT3 / DIN3	DOUT3 / DIN3	DOUT3 / DIN3	-
4	RS232 Rx	CAN L	-	-	-	-	-
5	RS232 Tx	CAN H	-	-	-	-	-
6	-	-	ENCA X	DOUT2 / DIN2	TTL Rx	DOUT2 / DIN2	-
7	-	-	ENCB X	Multi XY / X / Y	TTL Tx	DOUT1 / DIN1	-
8	-	-	-	-	-	-	Gnd

Optical Motion Detection (Optical Flow)

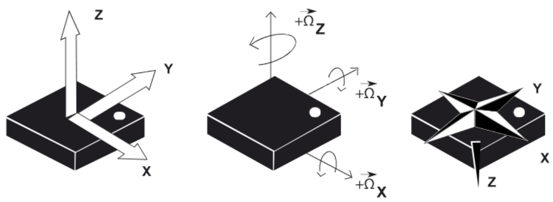
The FLW optical flow sensing is capable of tracking X-Y travelled distance on any kind of surface using a Laser for glossy or an LED for diffused surfaces.

It operates like an optical mouse but for higher distance, with higher precision, detecting and measuring the image translation while the sensor is moving.

Command	Arguments	Description	Example	Return
		Query Commands		
?C uu	uu = 0, 1, 2	Return raw count measurements on X and Y	?C 0 ?C 1 ?C 2	C=X:Y C=X C=Y
?T uu	uu = 0, 1, 2	Return raw delta count measurements on X and Y	?T 0 ?T 1 ?T 2	T=X:Y T=X T=Y
?MM uu	uu = 0, 1, 2	Return count measurements in .1 mm	?MM 0 ?MM 1 ?MM 2	MM=X:Y MM=X MM=Y
?SMM uu	uu = 0, 1, 2	Return speed measurements in .1 mm/s	?SMM 0 ?SMM 1 ?SMM 2	SMM=X:Y SMM=X SMM=Y
?IMQ	-	Return the image quality of the sensor, Laser IMQ should be > 1800, LED IMQ should be > 1300	?IMQ	IMQ=6352
		Configuration Commands		
^MMOD uu	uu = 0, 1	Configure between Laser or LED source	^MMOD 1	+
		Maintenance Commands		
%ZERO	-	Set new point of origin by resetting the counters	%ZERO	+

Inertial Measurement Unit (IMU)

The inertial measurement unit consists of a Gyroscope, an Accelerometer and a Magnetometer. All 9-axis measurements are available to the user to be used in a personalized script or to be requested by other units like motor controllers, a host device etc.



Command	Arguments	Description	Example	Return
		Query Commands		
?RMG uu	uu = 0, 1, 2, 3	Return gyroscope measurements	?RMG 0 ?RMG 1 ?RMG 2 ?RMG 3	RMG=GX:GY:GZ RMG=GX RMG=GY RMG=GZ
?RMA uu	uu = 0, 1, 2, 3	Return accelerometer measurements	?RMA 0 ?RMA 1 ?RMA 2 ?RMA 3	RMA=AX:AY:AZ RMA=AX RMA=AY RMA=AZ

Command	Arguments	Description	Example	Return
?RMM uu	uu = 0, 1, 2, 3	Return magnetometer measurements	?RMM 0 ?RMM 1 ?RMM 2 ?RMM 3	RMM=MX- :MY:MZ RMM=MX RMM=MY RMM=MZ
Configuration Commands				
^ZGYR uu	uu = 0, 1, 2, 3	Calibrate gyroscope X,Y and Z		+
^ZACC uu	uu = 0, 1, 2, 3	Calibrate accelerometer X,Y and Z		+
^ZMAG uu	uu = 0, 1, 2, 3	Calibrate magnetometer X,Y and Z		+
Maintenance Commands				
%GZER	-	Calibrate gyroscope	%GZER	+

Fusion Algorithm for Attitude Heading Reference System (AHRS)

The sensor utilizes a fusion algorithm to fully harness Attitude Heading Reference System capabilities, providing the user with orientation information in Euler Angles or Quaternions representation.

Real time 3D animation of the sensor's orientation is available in FLW PC Utility.

Command	Arguments	Description	Example	Return
Query Commands				
?EO uu	uu = 0, 1, 2, 3	Return AHRS output in roll, pitch and yaw	?EO 0 ?EO 1 ?EO 2 ?EO 3	EO=R:P:Y EO=ROLL EO=PITCH EO=YAW
?QO uu	uu = 0, 1, 2, 3, 4	Return AHRS output in quaternions	?QO 0 ?QO 1 ?QO 2 ?QO 3 ?QO 4	QO=A:B:C:D QO=A QO=B QO=C QO=D
Configuration Commands				
^EQS uu	uu = 0, 1	Configure between AHRS in Euler Angles (0) or Quaternions (1)	^EQS 0	+
^ZMT uu	uu = 0, 1, 2, 3	Configure AHRS fusion algorithm, Two options with or without magnetometer each Option 1 no mag (0), Option 1 (1), Option 2 no mag (2), Option 2 (3)	^ZMT 1	+
^KP uu	uu = 0,...,200	Configure proportional gain of fusion algorithm, User multiplier is 10, KP 100 represents a gain of 10.0	^KP 10	+
^KI uu	uu = 0,...,200	Configure integral gain of fusion algorithm, User multiplier is 10, KI 100 represents a gain of 10.0	^KI 0	+
Maintenance Commands				
%CLMOD uu	uu = 0, 1, 2, 3, 4	Calibrate IMU		+

Communication and Peripheral Multiplexers

The M-12 industrial type connector's I/Os are divided into two categories.
The first category would be titled as communication and the second as peripherals.
FLW offers a wide variety of connectivity options shared on two different busses and each bus may be configured to one of the corresponding categories.

Communication: CAN

Shared Peripherals: Quad Encoders, PWM, MultiPWM

Command	Arguments	Description	Example	Return
Configuration Commands				
^CCFG uu ww	uu = 2 ww = 0, 1, 2	Configure shared peripherals (uu = 2), Quad Encoders (ww=0) or Pulse (ww=1)	^CCFG 2 0	+

Built-In Scripting Language

In the following example, MicroBasic scripting language is used to extract a full sensor's image pixel by pixel.
This is may be used for investigating the sensor's image quality on various surfaces.

Note that during frame capture operation, the sensor will not be able to perform any other tasks, so this may not be used in real time applications.

Sample Script

```
#define ROW_SAMPLES   35
#define COL_SAMPLES   35
dim row as integer
dim col as integer
print("Start Frame Read\r")
print(«IR Source:\t\t\t»)
if (Getconfig(_MMOD,1) = 0) then
        print("Laser\r")
else
        print("LED\r")
end if
```

```
top:
Setcommand(_PXL, 0)
print("For Image Quality:\t\t",Getvalue(_IMQ,1),"\r")
   wait(100)
for row = 1 andwhile row <= ROW_SAMPLES
dump_row:
   print("PXL=")
   for col = 1 andwhile col <= COL_SAMPLES
         print(Getvalue(_PXL, col*row))
         if col <> COL_SAMPLES then
print(":")
   wait(3)
   next
   print("\r")
next
terminate
```

Command	Arguments	Description	Example	Return
!R uu	uu = 0, 1, 2	Control script, Pause script (0), run script (1), restart script (2)	!R 1	+

Communication and Peripherals

Communication

USB

Simple plug-n-play solution for monitoring, configuring and controlling the FLW sensor via RoboteQ's especially designed PC Utility.

CANbus

FLW sensor supports a multiprotocol CANbus implementation with four operating modes available.

1. CANOpen
2. MiniCAN
3. RawCAN
4. RoboCAN

RawCAN is a low-level operating mode giving total read and write access to CAN frames. It is recommended for use in low data rate systems that do not obey to any specific standard. CAN frames are typically built and decoded using the MicroBasic scripting language.

MiniCAN is greatly simplified subset of CANopen, allowing, within limits, the integration of the controller into an existing CANopen network. This mode requires MicroBasic scripting to prepare and use the CAN data.

CANopen is the full Standard from CAN in Automation (CIA), based on the DS302 specification. It is the mode to use if full compliance with the CANopen standard is a primary requisite.

RoboCAN is a Roboteq proprietary meshed networking scheme allowing multiple Roboteq devices to operate together as a single system. This protocol is extremely simple and lean, yet practically limitless in its abilities. It is the preferred protocol to use by users who just wish to make multiple controllers work together with the minimal effort.

Detailed descriptions of CANopen, MiniCAN, RawCAN and RoboCAN can be found in specific sections of RoboteQ's motor controller user manual.

MiniCAN Frames

TPDOs

Header:
TPDO1: 0x180 + NodeID
TPDO2: 0x280 + NodeID
TPDO3: 0x380 + NodeID
TPDO4: 0x480 + NodeID

Data:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
TPDO1	dX	dY	Status					
TPDO2	Counts X				Counts Y			
TPDO3	VAR1				VAR2			
TPDO4	BVar 1-8	BVar 9-16	BVar 17-24	BVar 25-32				

The Status byte transmits useful information about the sensor's current status in bit band representation.

To extract the status information from the status byte we need first to convert it into binary.

Then the data is the following:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor Fault	Run Script	LED Enabled	Laser Enabled		ToF Active	AHRS Active	Flow Active

RPDOs

Header:

RPDO1: 0x200 + NodeID
RPDO2: 0x300 + NodeID
RPDO3: 0x400 + NodeID
RPDO4: 0x500 + NodeID

Data:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
RPDO1	VAR3				VAR4			
RPDO2	VAR5				VAR6			
RPDO3	VAR7				VAR8			
RPDO4	BVar 33-40	BVar 41-48	BVar 49-56	BVar 57-64				

Command	Arguments	Description	Example	Return
Configuration Commands				
^CEN uu	CAN Mode Option	Disabled (0), CANOpen (1), MiniCAN (2), RawCAN (3), RoboCAN(4)	^CEN 0	+
^CNOD uu	Address	CAN Node Address	^CNOD 6	+
^CBR uu	Bit Rate Option	CAN Bit Rate, 1000k (0), 800k (1), 500k (2), 250k (3), 125k (4)	^CBR 3	+
^CHB	Heartbeat ms	CAN HeartBeat	^CHB 100	+
^CAS		CAN PDO Autostart	^CAS 1	+
^CLSN	Listen node	CAN Listening COB, Disabled (0), Nodes (1-127)	^CLSN 4	+
^CSRT	Sendrate ms	CAN SendRate	^CSRT 100	+
^CTPS	TPDO number, Sendrate ms	CAN TPDO Sendrate, If Sendrate = 0 TPDO is not transmitted	^CTPS 1 100	+
^CTT		CAN Open Type		+

Shared Peripherals

Quad Encoders

Quadrature encoder emulation is available in FLW sensor. This feature is enabled by default and it is one of the shared peripheral options.

The function is simple, for every +1 count, encoder A and B logical levels change according to the next state.

For every -1 count encoder A and B logical levels change according to the previous state.
If no counts are measured, A and B logical levels will stay the same.
From State 3 with a +1 count we are moving back to State 0 and from State 0 with a -1 count we are jumping to State 3.

By using the 4 encoder pins the sensor's X-Y displacement may be monitored by measuring the State transitions on ENC X pair and ENC Y pair.

Although FLW sensor has a .1 mm precision, the encoders by default will be trigger every 10 counts (every +/-1 mm). The prescalers may be configured as /1, /10, /100 and /1000 and the pulse width is fixed at 250 us.

By default (with 1 State transition every +/- 1mm), the maximum counts that may be transmitted continuously are 4000 counts/s.

Each t state in the following table is always 250 us.

	State 0	State 1	State 2	State 3
ENC A	1	1	0	0
ENC B	0	1	1	0

Example: Lets assume we are at State 1. That means that A and B are logically High. In case a +1 count need to be transmitted the ENC A and ENC B need to transit to State 2, which means A will become logically Low and B will stay logically High. If the next count is -1 we move back from State 2 to State 1 and both A and B are now logically High.

Command	Arguments	Description	Example	Return
Configuration Commands				
^EPPR uu ww	uu = 1 ww = 0, 1, 2, 3	Configure encoder prescaler, One state transition every .1mm (ww = 0), 1mm (ww = 1), 10mm (ww = 2) or 100mm (ww = 3)	^EPPR 1 1	+

MultiPWM

When Shared Peripherals option is configured in Pulse mode Quadrature Encoders, Serial TTL and DOUT1 are not available.

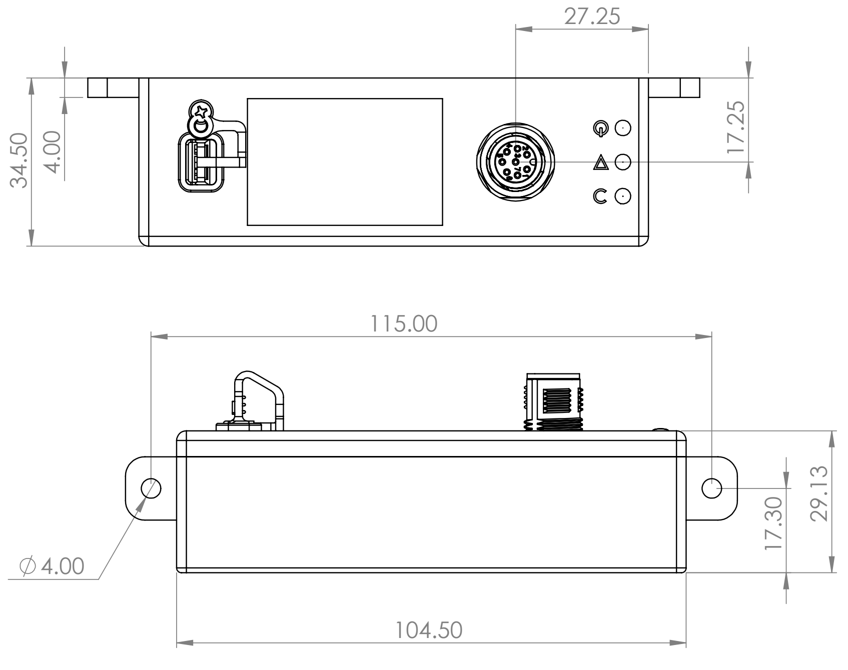
As the name implies, this proprietary method uses a succession of variable duty-cycle pulses to carry the measured X-Y distance and sensor's Status information to RoboteQ's controllers and extenders.

Connected on a common GND, all data are transmitted via a single wire interface.

MultiPWM may be enabled at Shared Peripheral options.

Command	Arguments	Description	Example	Return
Configuration Commands				
^PWMM uu	uu = 0, 1, 2	Configure pulse mode between MultiPWM XY (0), MultiPWM X (1) and MultiPWM Y (2)	^PWMM 0	+

Mechanical Dimensions



Appendix

FLW Configuration Summary

Sensor Configuration

FLWxxx is highly configurable offering a variety of options.

Configuration is optional but recommended for more personalized implementations.

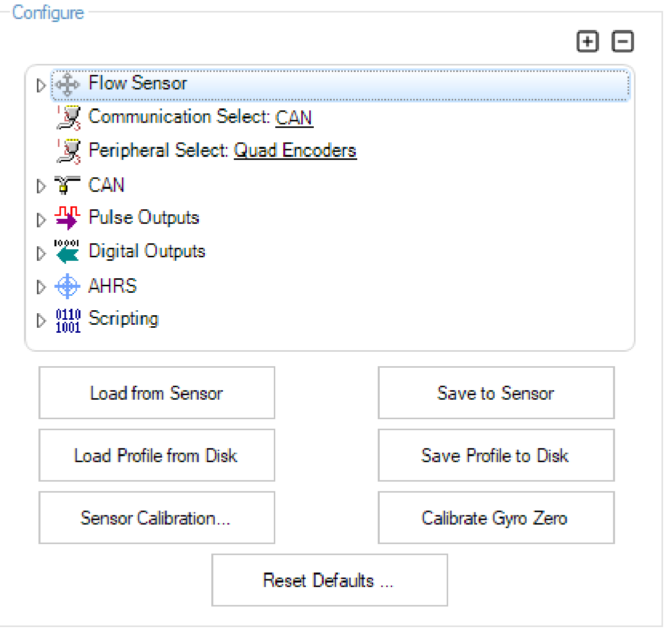
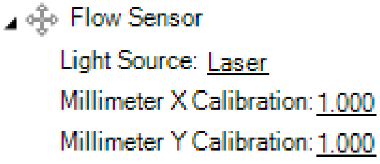


Image Flow Configuration

Light Source: It has two options Laser or LED. Both options refer to IR light with the former being used in glossy surfaces and the latter in diffused surfaces.

X and Y Calibrations: The FLWxxx sensor automatically converts the image flow into millimeters. In case the user wants to adjust or calibrate their measurements the easiest way is by using the calibration parameters.

Example: If the sensor measures 100 mm at X and the X calibration is 1.500, the exported Xmm value will be 100*1.500 = 150mm.

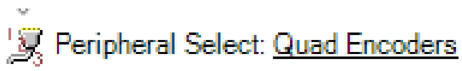


Shared Communication Select

Communication Select: Choose between RS232, RS485 and CAN.

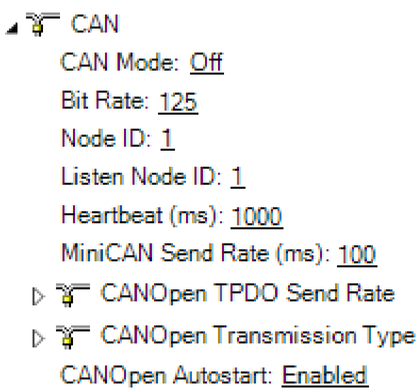
Shared Peripheral Select

Peripheral Select: Choose between Quad Encoders, Pulse, Serial TTL and Outputs Only options.



CAN Configuration

CAN Mode: Like in all RoboteQ's products 4 can protocols are supported. CANOpen, MiniCAN, RawCAN and RoboCAN.

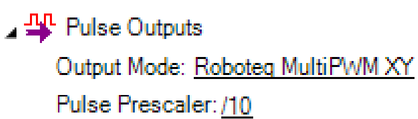


Pulse Configuration

Only if Pulse option has been chosen in Peripheral Select these configurations will apply.

Output Mode: Choose between Roboteq MultiPWM XY, Roboteq MultiPWM X and Roboteq MultiPWM Y.

Pulse Prescaler: The sensor's speed may be monitored by measuring the duty cycle of the pulses in PWM mode. The prescaler will reduce the precision of the transmitted speed for allowing higher speed values to be monitored (not available in FW v1.1 and below).

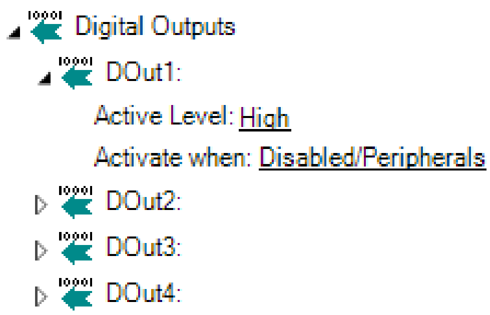


Digital Outputs

For these configurations to apply the corresponding Digital Out should not be reserved by any other operation (ex. Quad Encoders).

Active Level: This defined the active state of the Digital Output. When "High" the Digital Out is normally set at 0 V and rises to 5 V when activated. For a level "Low" the inverse applies.

Activate When: Choose between Disabled/Peripheral, User Out, SS1 Heartbeat. When Disabled/Peripheral option is used the Digital Output may be used as a Digital Input (if a peripheral utilizes this Digital Out the peripheral's outputs will be read in the Input). When User Out option is set the corresponding connector pin may be used to drive the gate of a digital switch operating in 0-5 V.

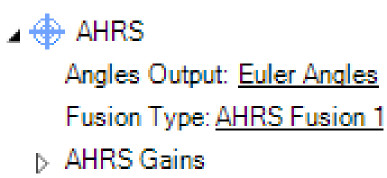


Attitude Heading Reference System Configuration

Angles Output: This option refers to the AHRS utility's tab animation. Euler Angles or Quaternions may be selected. Whichever the selection, both sets of variables are available for the user via "EO" (Euler) and "QO" (Quaternions) commands.

Fusion Type: One of two slightly different fusion algorithms may be chosen for the AHRS. Both options may be used as 9-Axis fusion or no Magnetometer fusion (6-Axis). The latter option will be useful in case external magnetic fields, like magnetic tapes, interfere with the sensor.

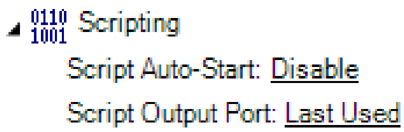
AHRS Gains: A proportional and an integral gain may be used for tuning the fusion algorithm's convergence.



Script Configuration

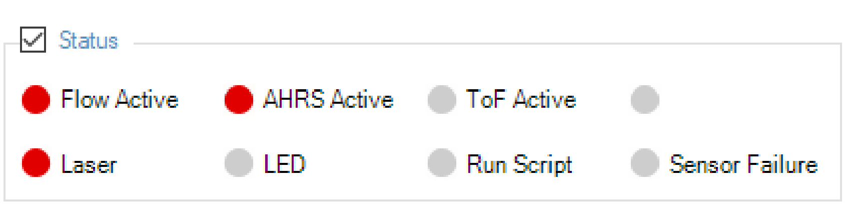
Script Auto-Start: This option might be set as either Enabled or Disabled. When Enabled, if there is a MicroBasic script loaded in the sensor, the script will start being executed after 2 seconds from sensor's power up.

Script Output Port: With this parameter all script exported data may be channeled to a specific communication peripheral. FLWxxx supports USB as an option for this parameter.



FLW Run

RoboteQ's FLW pc utility Run may be used for real time monitoring of the sensor.



Status

In this section all sensor status flags are displayed.

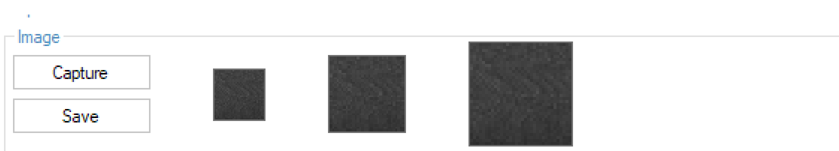
Flow Active, AHRS Active: The two main ICs for the FLW to operate properly are the Flow and the AHRS. When initialized correctly each flag will be set.

Sensor Failure: In case at least one of the Flow Active and AHRS Active fails to be initialized the Sensor Failure flag will be triggered.

Laser, LED: Laser or LED flag (not both) will always be set, indicating the IR light source used by the sensor to keep track of XY displacement.

Image

Capture: For debugging purposes the image capture feature may be used. This is not a real time feature, meaning that the sensor will stall all operations while dumping an image frame, but it could be used to observe the actual image the sensor is capturing over a specific surface.



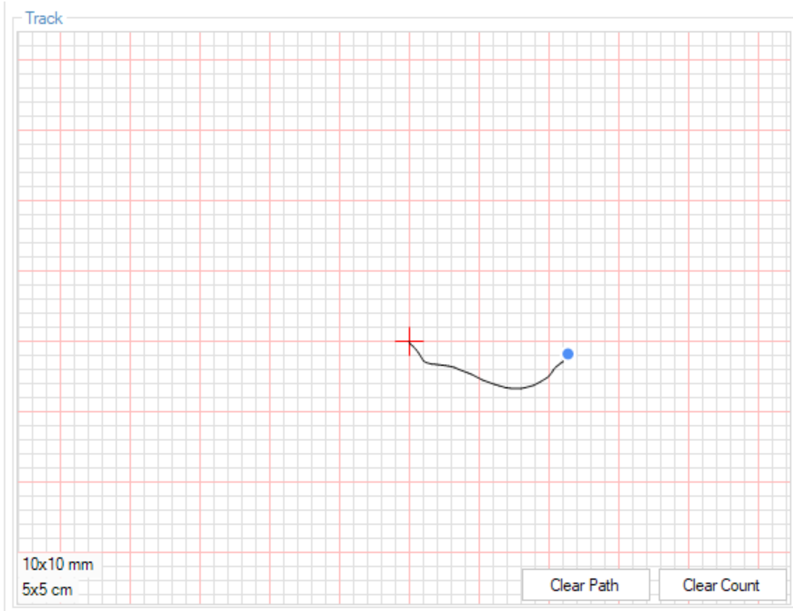
Save: By using the Save button a png file of the captured image will be generated for future processing.

Track

In Track section the real time X-Y displacement of the sensor is captured.

Clear Path: This button clears all captured trace of the FLW sensor. The sensor's current position remains the same.

Clear Count: The Clear Count button is equivalent of sending %ZERO command via console, it resets the X and Y counts to zero. The captured trace is not cleared on the grid when using this button.



Capture Graph

Many FLW variables are available for plotting and logging.

Record/Pause: When Record is chosen all configured variables are being captured on the graph. Pause could be pressed for the real time plotting to be paused.

Save: Using the Save button a log will be extracted and be saved for future analysis.

Clear Chart: All captured variables will be cleared from the chart. This will not affect the logs already captured .

Clear Log: This will erase all logs taken until now allowing the user to set a new point of reference for logging. None of the plotted variables will be erased from the graph.

Log Configuration: This feature allows the user to store channel configurations (colors and variables) for future use.