Physical Setup

Listed below are the different functions for each wire contained in the Mainprobe-V RS485 cable.

Insulation colour	Purpose	Mainstream RS485 cable colour
Yellow	RS485 Half Duplex B-	Yellow
Brown	RS485 Half Duplex A+	Orange
Red	Power (+ve)	
Black	Power (-ve)	
Blue	Reserved/Not fitted	
White	Reserved/Not fitted	
Clear	Screen	

Input voltage (1 device on bus): 6V – 28V   **Input voltage (>1 device on bus): 7-28V**

Screen connection: Should be connected to 0V or terminated.

Line termination: See official Modbus documentation.

Additional parameters concerning the physical interface to the product can be found in the user manual..
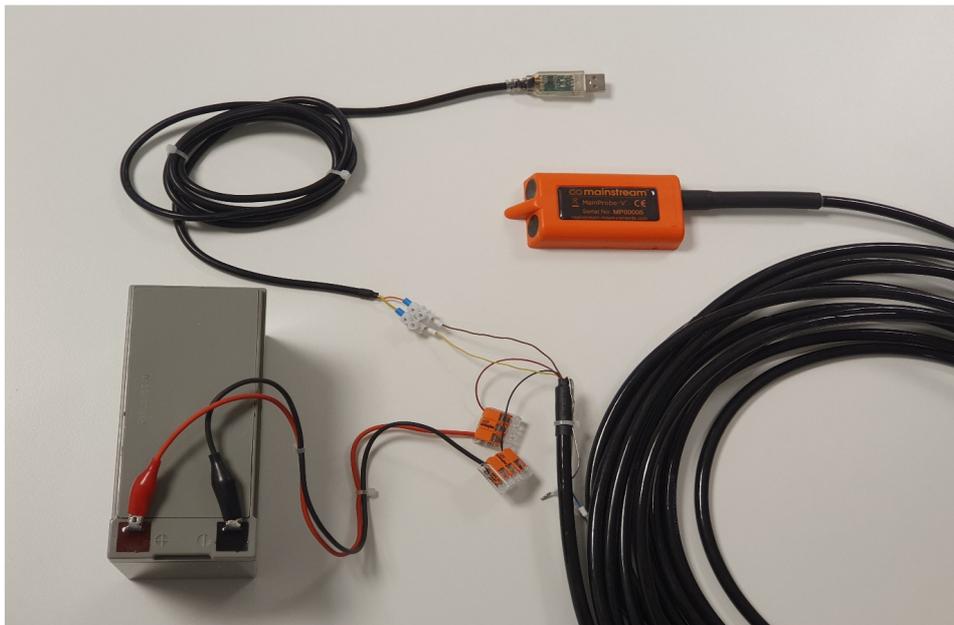
As always when dealing with electrical connections, care must be taken to avoid short circuits etc when wiring up the MainProbe to the power supply.

How to interface with the MainProbe

The MainProbe can be interfaced with using one of two methods:

       1: Using the MainProbe Communicator software.

       2: Using an RS485 half-duplex Modbus-RTU client device.

Interfacing using MainProbe Communicator

MainProbe Communicator is Windows software which allows the MainProbe to be configured and managed. MainProbe Communicator acts as a Modbus client device, thus, no other Modbus client can be active on the bus while the MainProbe Communicator is being used.

1:   Installing Communicator

Minimum system requirements: Windows 10, .NET 5.0.1(*) desktop runtime SDK.

(*) Or better. Some windows 10 PC's will have the .NET 5 desktop runtime SDK installed already. Some may require manual installation.

Any USB drive provided by Mainstream or "MainProbe Communicator.zip" file, will a include a setup file for Communicator. This setup file will install Communicator in a folder of the user's choice.

2:   .NET 5.x

If you install Communicator and try to run the software without the .NET 5.x installed, you will be shown an error message, prompting you to install the .NET runtime, clicking yes on this message will direct you to a Microsoft webpage where .NET can be downloaded. At time of writing this is https://dotnet.microsoft.com/download/dotnet/5.0 . Ensure you download the **x64** SDK.

You can now run Communicator.

Accessing the MainProbe:

1: Ensure the MainProbe is connected to the provided RS485 to USB adapter cable correctly and the device is powered.

2: To communicate with the MainProbe you must know the current **communications settings**. Ensure these settings are selected in the left-hand pane of Communicator. By default settings are as follows:

Baud rate: 19200, Parity: Even, Stop bits: 1, Modbus Address: 1

The correct COM port can be found by navigating to the windows "device manager", expanding the "Ports (COM & LPT) section, un-plugging and plugging-in the USB adapter should identify the COM port being used to host the RS485 to USB adapter.

3: Click the **Scan** button to detect the MainProbe, if multiple devices are connected to the RS485 bus, they can be scanned for simultaneously using a hyphen in the Modbus Address box, (e.g., 1-3, would scan address's 1, 2 and 3).

4: Once the scan is complete (shown right), a list of thumbnails detailing the detected devices will be presented. If the device that you wish to connect to is present, click the **Connect** button.
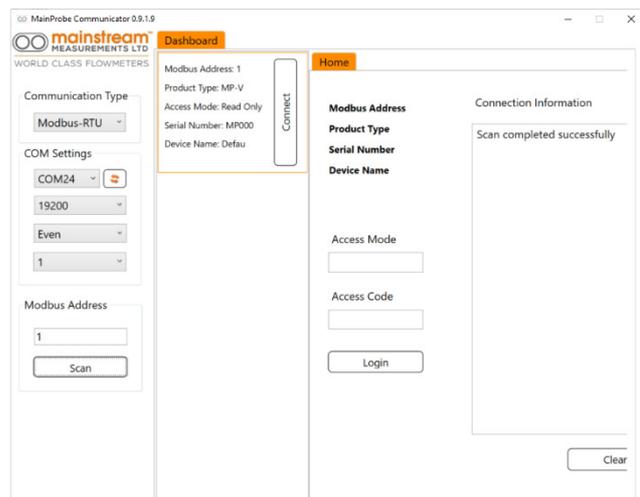
Once connection is successful, you are now in a session with the MainProbe.

Upon power-up the device will be in read-only mode. Other access modes can be accessed by entering one of the 4-digit codes listed below followed by clicking the **Login** button.

Read-Only: 0000 - Used to force measurements, edit measurement units, and view the device configuration.

Basic: 1234 - Same as Read-Only mode + used to alter communications settings, custom device name, measurement interval and restore default settings.

Advanced: 5678 – Same as basic mode + used to modify all other configurable settings.
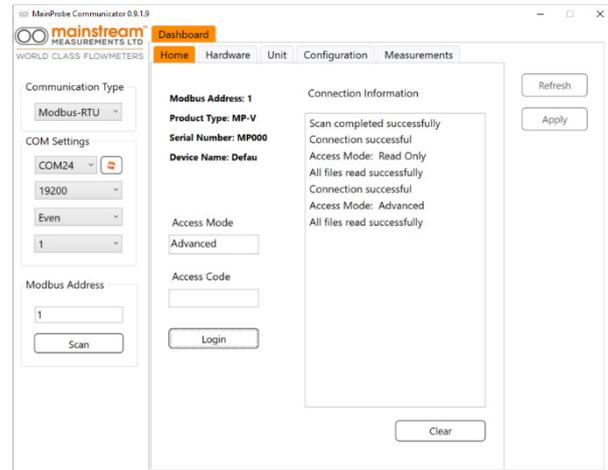
An incorrect password, closing the software or power-cycling the probe will reset the MainProbe to Read-Only mode. Shown above is a successful login to a unit in Advanced mode.

Sensor Configuration:

Once you have logged into the unit (if necessary), any changes you make will need to be written to the MainProbe, this can be done using the **Apply** button.

Clicking the **Refresh** button will update the information shown on the current tab.

The key setting for initial sensor configuration is the **Measurement Interval**. Setting this to "On Demand" will result in the sensor only taking readings when the user manually forces a measurement.

Alternatively, the sensor can be configured to take regular readings.

Once you have configured the device, verify your installation by viewing the Measurements tab.

You can manually invoke a measurement using the **Force Measurements** button.

Further details on how to interpret the velocity signals and velocity histogram can be found in the MainProbe User Guide.

(Advanced mode only)

Clicking the **Restore Defaults** button will restore the defaults for the current tab and any subsequent sub-tabs (e.g., clicking this button on the Configuration -> Modbus RTU sub-tab, will also restore the settings on the Configuration -> Velocity Sensor sub-tab).
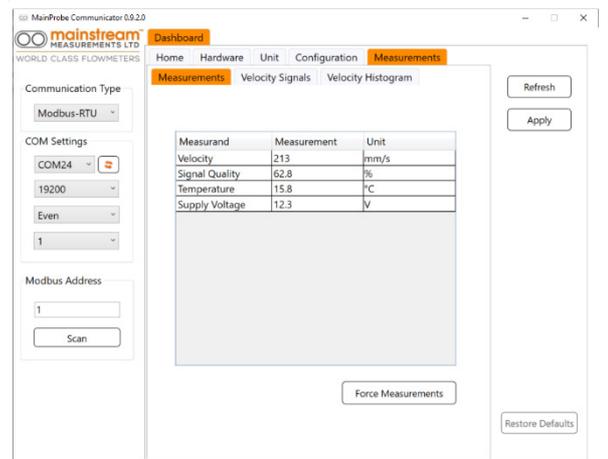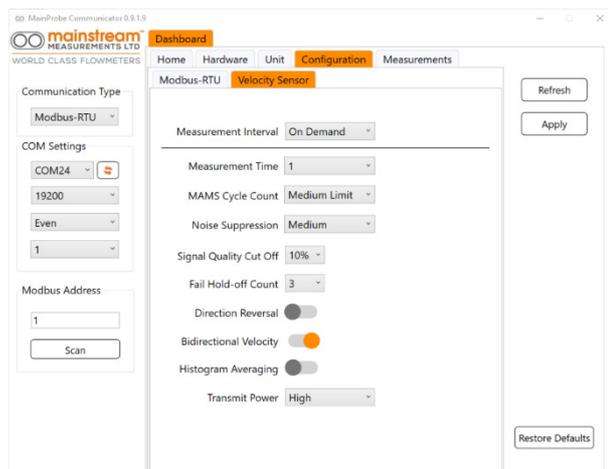
Interfacing using a Modbus client

Supported Modbus function codes

All data in the MainProbe is stored in **16-bit holding registers**, these can be accessed using the following function codes.

3 – Read holding register(s)

6 – Write holding register

Note that only one holding register can be written to, per Modbus message. Function code 16 (write multiple holding registers) is not available.

Supported Modbus exception codes

Exception code 1 - Illegal Function.

Exception code 2 - Illegal Data Address.

Exception code 3 - Illegal Data value - Exception code 3 is used to help express violations to the permissions system.

Supported Data Types

Byte arrangement is fixed to a **low word first, high byte first** configuration.

This means that for every 16-bit word requested from the device, of the two bytes received, the **most significant byte** will be sent **first**.

When more than one 16-bit register is requested, these 16-bit words are arranged into 32-bit values. Within each 32-bit value, the **pairs of words** are sent in reverse order. So, the **least significant word**, is sent **first** (this is because the lease significant word holds the lowest value Modbus address, thus the). This has been set in accordance with the demands of most Modbus client requirements.

float- Constructed using 2, concatenated 16-bit registers:

The floating-point number:	"212.549530"	**(ABCD)**
Can be expressed as the bytes:	"43 54 8C AE"	**(ABCD)**
These will be received from the MainProbe as:	"8C AE 43 54"	**(CDAB)**
The register addresses for this data would be:	"8C AE = 4xxxx"	**(CD)**
	"43 54 = 4xxxx+1"	**(AB)**

int32 – Constructed using 2, concatenated 16-bit registers:

The 32-bit integer decimal number:	"305419896"	**(ABCD)**
Can be expressed as the bytes:	"12 34 56 78"	**(ABCD)**
These will be received from the MainProbe as:	"56 78 12 34"	**(CDAB)**
The register addresses for this data would be:	"12 34 = 4xxxx"	**(CD)**
	"56 78 = 4xxxx+1"	**(AB)**

Many of the registers in the device are listed as 32-bit integers, many of them will never reach values which require more than 16 bits to express, for these registers it is simpler to only read the least significant 16-bits of these 32-bit values.

Start-up sequence and latencies:

The MainProbe has a **start-up time of ~60ms**. At this point requests can be sent to the product.

The MainProbe will immediately begin taking a measurement upon start-up.

Each second of sampling (indicated by "Measurement Time" parameter), requires ~400ms of additional processing time.

Supported Modbus holding registers

To extract a 32-bit data types, 2 holding registers should be requested.

The Mainprobe-V RS485 features an access mode system which allows access to be granted and revoked to different parameters on the product, to allow varying types of users to operate the product, without the risk of misconfiguration. The permissions are assigned to each parameter and listed below. For more information on this system, see the MainProbe user manual.

A full register list can be found in the user guide.

Parameter Name	Modbus Register Address / Reg. Count	Description / Data type / Default value /	Access Mode		
			Adv.	Basic	Read Only
Configuration File					
Access Mode	41132 / 2	0 = Read-Only mode, 1 = Basic mode, 2 = Advanced mode. Data type: Int32 / Default value: 0 (Read-Only Mode)	Rd. Only	Rd. Only	Rd. Only
Access Code	41134 / 2	Write a correct password to the lower word of this register to login. Password should be in numerical hex. For basic mode, write 0x1234 to register 41096, no need to write to 41097. Data type: Int32 / Default value: Parameter will always read as 0x50415353.	Rd. Wr.	Rd. Wr.	Rd. Wr.
Baud Rate *	41136 / 2	0=2400, 1=4800, 2=9600, 3=19200, 4=38400, 5=57600, 6=115200 **See "Update Communications"** Data type: Int32 / Default value: 3 (19200 Baud).	Rd. Wr	Rd. Wr.	Rd. Only
Stop Bits *	41138 / 2	1=1 Stop bit, 2=2 Stop bits. **See "Update Communications"** Data type: Int32 / Default value: 1 (1 Stop bit).	Rd. Wr.	Rd. Wr.	Rd. Only
Data Bits *	41140 / 2	7=7 Data bits, 8=8 Data bits. **See "Update Communications"** Data type: Int32 / Default value: 8 (8 Data bits).	Rd. Only	Rd. Only	Rd. Only
Parity *	41142 / 2	0=No Parity, 1=Odd Parity, 2=Even Parity. **See "Update Communications"** Data type: Int32 / Default value: 2 (Even parity).	Rd. Wr.	Rd. Wr.	Rd. Only
Modbus Server Address	41144 / 2	Can be set to any valid Modbus client address 1-247 inclusive. Data type: Int32 / Default value: 1	Rd. Wr.	Rd. Wr.	Rd. Only
Update Communications	41146 / 2	Registers marked with a * can be written to but the changes will not take effect until a 1 is written to this register or the device is power cycled. Data type: Int32 / Default value: 1	Rd. Wr.	Rd. Wr.	Rd. Only
Measurement Time	41156 / 2	1-10 Available. Represents the duration of measurement taken, in seconds, does not include processing time. Data type: Int32 / Default value: 1 (1 Second).	Rd. Wr.	Rd. Only	Rd. Only
Measurement Interval	41158 / 2	0=On Demand, 1=15s, 2=30s, 3=1min, 4=2mins, 5=5mins, 6=10mins, 7=15mins, 8=20mins, 9=30mins, 10=60mins. **See "Taking a measurement" before writing to this register.** Data type: Int 32 / Default value 2 (30 Seconds).	Rd. Wr.	Rd. Wr.	Rd. Only
Measurements File					
Measurements Complete	41374 / 2	1=Not in progress, 0=In progress. Indicates whether a set of measurements is currently in progress. **See "Taking a measurement" before writing to this register.** Data type: Int32 / Default value: 1 (Measurements complete).	Rd. Wr.	Rd. Wr.	Rd. Wr.
Velocity (mm/s)	41376 / 2	Returns the most recent velocity measurement in mm/s. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Velocity (cm/s)	41378 / 2	Returns the most recent velocity measurement in cm/s. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only

Velocity (M/s)	41380 / 2	Returns the most recent velocity measurement in M/s. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Velocity (in/s)	41382 / 2	Returns the most recent velocity measurement in in/s. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Velocity (ft/s)	41384 / 2	Returns the most recent velocity measurement in ft/s. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Velocity (ft/min)	41386 / 2	Returns the most recent velocity measurement in ft/min. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Signal quality (%)	41388 / 2	Returns the signal quality of the most recent velocity measurement in %. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Temperature (°C)	41390 / 2	Returns the most recent temperature reading in degrees Celsius. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Temperature (°F)	41392 / 2	Returns the most recent temperature reading in degrees Fahrenheit. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only
Supply Voltage (V)	41394 / 2	Returns the most recent supply voltage reading in Volts. Data type: Float / Default value: 0.0	Rd. Only	Rd. Only	Rd. Only

Taking a measurement

There are two possible trigger sources that can result in the MainProbe taking a measurement **(velocity, signal quality, temperature and supply voltage)**.

> **1: The measurement interval timer elapsing.**

> **2: Writing a 0 to the "Measurements Complete" register.**

When set, every time the Measurement Interval elapses, a measurement is taken.

When a set of measurements is taken, the "Measurements Complete" register is set to a **0**, it can be used as an indicator to determine when a measurement has been completed.

Writing a **0** to the "Measurements Complete" register will force the device to take a measurement, provided the device is not already busy taking a measurement.

If the Measurements Complete register is currently a 0 (measurements in progress), and the user tries to set the register to a 0, no additional measurement will be forced or queued. The user should check that the Measurements Complete register is set to a 1 (no measurement in progress) before attempting to trigger a measurement.

Similarly, if the user manually invokes a measurement using the Measurements Complete register and the measurement interval timer then elapses, triggering another measurement. This second, interval triggered, measurement will not be performed, and the device will wait for the next trigger to occur.

The procedure for manual measurement taking should be as follows:

1: Check the Measurements Complete register is set to a 1 (device not currently busy).

2: Write a 0 to the Measurements Complete register.

3: Read the Measurements Complete register periodically (1s intervals), until the register reads back a 1, velocity should not be read.

4: If the measurement time is known to the user, you can reduce bus accesses by waiting for at least this duration of time to elapse before re-checking the Measurements Complete register.

5: Once the measurements are complete. The user can read the desired velocity register depending on their preferred unit.

For standard applications where a velocity reading is taken regularly, it is simpler to set the MainProbe to a preset measurement interval and poll velocity at a regular interval controlled by the Modbus master device. The previously described sequence is only necessary where longer or abnormal durations are required between measurements and the user does not wish to waste any power taking additional measurements. Note that during high measurement times with short intervals, the device may be busy more often than it is not, responses slow. If it can be avoided, it is better to avoid sending large data requests to the device during a measurement.