6. Modbus TCP

From firmware version 4.0.0, the Asycube offers a Modbus TCP slave (=server) interface in order to simplify the integration with PLCs and other Modbus TCP masters (=clients) over TCP/IP networks. More information on Modbus can be found in the protocol specification available on http://www.modbus.org.

The Section 6.1 presents all necessary information to communicate with the Asycube through Modbus TCP. The Section 6.2 shows typical performances that can be achieved. The Asycube Modbus Register Table is presented in Section 6.3. The Section 6.4 explains the different types of errors and the way to handle them. Finally, the Section 6.5 gives a few useful examples, such as how to start a platform vibration through Modbus TCP.

6.1. Configuration

Default Modbus TCP parameters for the Asycube are:

IP Address	192.168.127.254
Subnet Mask	255.255.255.0
Modbus port	502
Unit ID	N/A for Modbus TCP

The IP Address and the Subnet Mask are shared with the Ethernet communication (Section5.1) and can be changed in the configuration page of the Asyril HMI unlike the Modbus port, which is fixed to 502.

Out of all Modbus public function codes, the Asycube implements the minimum useful set of functions (Class 0):

- Code 03 (0x03): Read Holding Registers, with 16 bits access
- Code 16 (0x10): Write Multiple Registers, with 16 bits access

6.2. Performance

This section presents the performance and limitations of the Asycube Modbus TCP implementation.

6.2.1. Communication

The Asycube only accepts one Modbus master/client to be connected at a time. However, a TCP/IP connection (See Chapter 5) can be used in parallel with Modbus; this can be useful to use the Asycube HMI to configure the Asycube while a machine PLC controls the Asycube through Modbus.

The Asycube can only handle one read or write request at a time. If two requests are sent simultaneously, the second one will be ignored. It is recommended to alternate between the different read/write requests.

As stated in the Modbus protocol specification, up to 125 registers can be read at once and only up to 123 registers can be written at once. If more registers need to be read (or written), you will have to implement multiple read (or write) requests one after the other.

6.2.2. Timing

The Asycube is able to handle one Modbus read or write request every 4 ms. The timing performance of a Modbus command (e.g. trig the start of a platform vibration) depends on different factors such as the Modbus master implementation or the Ethernet network state. Therefore, it is not possible to guarantee neither real-time behavior nor maximal reaction time. The order of magnitude for the delay between a Modbus command and its effect on the Asycube is 10 ms. The delay between the instant when a register is written and the instant when its updated value is read on Modbus typically lays around 15 ms.

6.3. Asycube Modbus Register Table

Asycube data can be accessed via Modbus TCP through Holding Registers. The Register Table presented in this section describes all the Asycube Holding Registers. All Holding Registers are 2 bytes long. Their type is generally a WORD or an UNSIGNED_INT16. Some of the Holding Registers can have negative values (see Data Range column in the Register Table). In this case, their type is SIGNED_INT16. All registers are either Read-only or Write-only (see Read/Write column in the Register Table). This section presents the Holding Registers, which have been placed in different address zones. The registers addresses are shown with the address offset of the zone and the relative address of the register (e.g. “64+3” means that the register is in the zone starting at address 64).

6.3.1. Control Holding Registers (Write-only zone, offset=0)

The Control Holding Registers presented in Table 6-1 are used to control the Asycube. Examples: start a vibration, start a sequence, clear the errors, …

The Holding Registers highlighted in **bold** are triggers that start an action or change a value.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
0	HR_MODBUS_CONTROL	W	0 to 15	The 4 first bits are used to clear the errors. A rising edge on a bit clears the corresponding error. More information in Section 6.4.
1	**HR_PLATFORM_VIBRATION_TRIG**	W	0 or 1	A rising edge trigs the start of a platform vibration with the parameters values given in HR_PLATFORM_VIBRATION_ID and HR_PLATFORM_VIBRATION_DURATION.
2	HR_PLATFORM_VIBRATION_ID	W	0 to 25	The platform vibration identifier is used when a platform vibration is trigged with HR_PLATFORM_VIBRATION_TRIG. The value 0 corresponds to the vibration 'A', the value 1 corresponds to the vibration 'B', …, the value 25 corresponds to the vibration 'Z'. Please refer to Section 2.1 for the conventions of direction.
3	HR_PLATFORM_VIBRATION_DURATION	W	0 to 30000 [ms]	This value lets the user chose the duration of the platform vibration when it is trigged with HR_PLATFORM_VIBRATION_TRIG.
4	**HR_PLATFORM_CENTERING_TRIG**	W	0 or 1	A rising edge (value change from 0 to 1) trigs the start of a platform centering with the parameters values given in HR_PLATFORM_CENTERING_X and HR_PLATFORM_CENTERING_Y. More information in Section 5.4.10.2.
5	HR_PLATFORM_CENTERING_X	W	-100 to 100	The platform centering trigged with HR_PLATFORM_CENTERING_TRIG uses this X position to determine both the vibration direction and the duration. This value is an integer and corresponds to 100x the value described in the

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
				coordinate system (Section 3.3.6); a value of -60 in this Holding Register corresponds to -0.6 in the defined coordinate system.
6	HR_PLATFORM_CENTERING_Y	W	-100 to 100	The platform centering trigged with HR_PLATFORM_CENTERING_TRIG uses this Y position to determine both the vibration direction and the duration. This value is an integer and corresponds to 100x the value described in the coordinate system (Section 3.3.6); a value of -60 in this Holding Register corresponds to -0.6 in the defined coordinate system.
7	**HR_HOPPER_VIBRATION_TRIG**	W	0 or 1	A rising edge trigs the start of a hopper vibration with the parameters values given in HR_HOPPER_VIBRATION_ID and HR_HOPPER_VIBRATION_DURATION.
8	HR_HOPPER_VIBRATION_ID	W	0 to 25	The hopper vibration identifier is used when a hopper vibration is trigged with HR_HOPPER_VIBRATION_TRIG. The value 0 corresponds to the vibration 'A', the value 1 corresponds to the vibration 'B', …, the value 25 corresponds to the vibration 'Z'. Please refer to Section 2.1 for the conventions of direction.
9	HR_HOPPER_VIBRATION_DURATION	W	0 to 30000 [ms]	This value lets the user chose the duration of the hopper vibration when it is trigged with HR_PLATFORM_HOPPER_TRIG.
10	**HR_HOPPER_FEEDING_TRIG**	W	0 or 1	A rising edge trigs the start of a hopper feeding with the parameters values given in HR_HOPPER_FEEDING_VIBRATION, HR_HOPPER_FEEDING_NBPARTS and HR_HOPPER_FEEDING_NBMAX. More information in Section 5.4.10.3.
11	HR_HOPPER_FEEDING_VIBRATION	W	0 to 25	The hopper feeding trigged with HR_HOPPER_VIBRATION_TRIG uses this hopper vibration identifier. The value 0 corresponds to the vibration 'A', the value 1 corresponds to the vibration 'B', …, the value 25 corresponds to the vibration 'Z'. Please refer to Section 2.1 for the conventions of direction.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
12	HR_HOPPER_FEEDING_NBPARTS	W	0 to 65535	The hopper feeding trigged with HR_HOPPER_VIBRATION_TRIG uses this number of parts. More information in Section 5.4.10.3.
13	HR_HOPPER_FEEDING_NBMAX	W	0 to 65535	The hopper feeding trigged with HR_HOPPER_VIBRATION_TRIG uses this maximum number of parts. More information in Section 5.4.10.3.
14	**HR_SEQUENCE_EXECUTION_TRIG**	W	0 or 1	A rising edge trigs the start of a sequence execution with the parameters values given in HR_SEQUENCE_EXECUTION_NBPARTS, HR_SEQUENCE_EXECUTION_NBMAX, HR_SEQUENCE_EXECUTION_X, HR_SEQUENCE_EXECUTION_Y and HR_SEQUENCE_EXECUTION_SEQUENCEID. More information in Section 5.4.10.1.
15	HR_SEQUENCE_EXECUTION_NBPARTS	W	0 to 65535	The sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG uses this number of parts. More information in Section 5.4.10.1.
16	HR_SEQUENCE_EXECUTION_NBMAX	W	0 to 65535	The sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG uses this maximum number of parts. More information in Section 5.4.10.1.
17	HR_SEQUENCE_EXECUTION_X	W	-100 to 100	The sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG uses this X position to determine both the vibration direction and the duration. This value is an integer and corresponds to 100x the value described in the coordinate system (Section 3.3.6); a value of -60 in this Holding Register corresponds to -0.6 in the defined coordinate system.
18	HR_SEQUENCE_EXECUTION_Y	W	-100 to 100	The sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG uses this Y position to determine both the vibration direction and the duration. This value is an integer and corresponds to 100x the value described in the coordinate system (Section 3.3.6); a value of -60 in this Holding Register

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
				corresponds to -0.6 in the defined coordinate system.
19	HR_SEQUENCE_EXECUTION_SEQUENCEID	W	1 to 26	The sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG uses this sequence ID. More information in Section 5.4.10.1.
20	**HR_SELECT_VIBRATION_SET**	W	1 to 26	A value change of this Holding Register sets the new vibration set to use.
21	**HR_SELECT_SEQUENCE**	W	1 to 26	A value change of this Holding Register sets the new sequence to use.
22	**HR_BACKLIGHT_STATE_CONTROL**	W	0 or 1	A rising edge switches the backlight ON. A falling edge (value change from 1 to 0) switches the backlight OFF.
23	**HR_EXECUTE_BACKLIGHT_FLASH_TRIG**	W	0 or 1	A rising edge trigs a backlight flash. More information in Section 5.4.6.
24	**HR_BACKLIGHT_INTENSITY_TRIG**	W	0 or 1	A rising edge sets the backlight intensity value given in the Holding Register HR_BACKLIGHT_INTENSITY.
25	HR_BACKLIGHT_INTENSITY	W	*min* to 100 [%]	This backlight intensity can be set by trigging the HR_BACKLIGHT_INTENSITY Holding Register. The *min* value corresponds to the minimum intensity settable for the Asycube, which value can be read in the corresponding configuration parameter (Section 4.1).
26	**HR_PLATFORM_HALT**	W	0 or 1	A rising edge halts the platform vibration.
27	**HR_HOPPER_HALT**	W	0 or 1	A rising edge halts the hopper vibration.
28	**HR_SEQUENCE_HALT**	W	0 or 1	A rising edge halts the sequence execution.
29	**HR_READ_WRITE_PARAMETER_TRIG**	W	0 or 1	A rising edge reads or writes an Asycube parameter (Chapter 4) with the address given in the Holding Register HR_READ_WRITE_PARAMETER_ADDRESS. An even address trigs a read where the resulting value can be read in the Holding Register HR_READ_WRITE_PARAMETER_READ_VALUE (Address 64+13). An odd

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
				address trigs a write of the value given in the Holding Register HR_READ_WRITE_PARAMETER_WRITE_VALUE (Address 30). Note: no error occurs when trying to write in a forbidden Asycube register.
30	HR_READ_WRITE_PARAMETER_ADDRESS	W	0 to 65535	The parameter read/write trigged with HR_READ_WRITE_PARAMETER_TRIG needs this parameter address. More information in Chapter 4 and in the row above.
31	HR_READ_WRITE_PARAMETER_WRITE_VALUE	W	0 to 65535	The parameter write trigged with HR_READ_WRITE_PARAMTER_TRIG needs this Holding Register value to be written in the parameter.
32	**HR_FLASH_OPERATIONS**	W	0 to 15	The 4 first bits are used to start flash operations. A rising edge on a bit trigs the corresponding flash operation: 0b0001: Flash All in memory 0b0010: Flash Global Parameters 0b0100: Flash Sequences 0b1000: Flash Vibration Set More information in Section 5.4.11.

Table 6-1: Control Holding Registers, offset=0

6.3.2. Status Holding Registers (Read-only zone, offset=64)

The Status Holding Registers presented in Table 6-2 are used to get information on the Asycube. Examples: remaining vibration time, error state, …

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
64+0	HR_MODBUS_STATUS	R	0 to 15	The 4 first bits represents the errors status: Asycube Modbus errors status, Internal Modbus communication error status, Asycube warnings status and Asycube alarms status. More information in Section 6.4.
64+1	HR_MODBUS_ERROR_CODE	R	0 to 65535	The first byte (mask 0x0F) of this Holding Register contains Asycube Modbus error code, while the second byte (mask 0xF0) contains the Internal Modbus communication error code. More information in Section 6.4.2.
64+2	HR_WARNINGS	R	0 to 255	This Holding Register contains the Asycube warning code. More information in Section 6.4.3 and Section 5.4.14.
64+3	HR_ALARMS	R	0 to 255	This Holding Register contains the Asycube alarm code. More information in Section 6.4.3 and Section 5.4.15.
64+4	HR_MODBUS_PLATFORM_STATUS	R	0: DONE 1: BUSY 2: ERROR	This Holding Register gives the status of either the platform vibration trigged with HR_PLATFORM_VIBRATION_TRIG or the platform centering trigged with HR_PLATFORM_CENTERING_TRIG. The transition DONE->BUSY occurs on the trigger rising edge if the data is correct. The transition DONE->ERROR occurs on the trigger rising edge if the data is incorrect (e.g. index is out of range). The transition BUSY->DONE occurs as soon as the vibration ends. The transition BUSY->ERROR occurs if an Internal Modbus communication error (See Section 6.4.2) before the vibration starts.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
				The transition ERROR->DONE can be controlled by setting both triggers to 0.
64+5	HR_PLATFORM_REMAINING_TIME	R	0 to 32767 [ms]	This Holding Register gives the remaining time of the platform vibration. During a vibration, the value decreases over time until it reaches 0. Note: A value of 32767 means that the remaining time is greater than 32767ms. Note: Just before the vibration starts, during the amplifier start up, the value is different than 0 but can be strange. Similarly, during a continuous vibration, the value stays different than 0 but can be strange.
64+6	HR_MODBUS_HOPPER_STATUS	R	0: DONE 1: BUSY 2: ERROR	This Holding Register gives the status of either the hopper vibration trigged with HR_HOPPER_VIBRATION_TRIG or the hopper feeding trigged with HR_HOPPER_FEEDING_TRIG. The transition DONE->BUSY occurs on the trigger rising edge if the data is correct. The transition DONE->ERROR occurs on the trigger rising edge if the data is incorrect (e.g. index is out of range). The transition BUSY->DONE occurs as soon as the vibration ends. The transition BUSY->ERROR occurs if an Internal Modbus communication error (See Section 6.4.2) before the vibration starts. The transition ERROR->DONE can be controlled by setting both triggers to 0.
64+7	HR_HOPPER_REMAINING_TIME	R	0 to 32767 [ms]	This Holding Register gives the remaining time of the hopper vibration. During a vibration, the value decreases over time until it reaches 0. Note: A value of 32767 means that the remaining time is greater than 32767ms. Note: Just before the vibration starts, during the amplifier start up, the value is different than 0 but can be strange. Similarly, during a continuous vibration, the value stays different than 0 but can be strange.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
64+8	HR_MODBUS_SEQUENCE_STATUS	R	0: DONE 1: BUSY 2: ERROR	This Holding Register gives the status of the sequence execution trigged with HR_SEQUENCE_EXECUTION_TRIG. The transition DONE->BUSY occurs on the trigger rising edge if the data is correct. The transition DONE->ERROR occurs on the trigger rising edge if the data is incorrect (e.g. index is out of range). The transition BUSY->DONE occurs as soon as the sequence ends. The transition BUSY->ERROR occurs if an Internal Modbus communication error (See Section 6.4.2) before the sequence starts. The transition ERROR->DONE can be controlled by setting the trigger to 0.
64+9	HR_SEQUENCE_REMAINING_TIME	R	0 to 32767 [ms]	This Holding Register gives the remaining time of the sequence execution. During a sequence, the value decreases over time until it reaches 0. Note: A value of 32767 means that the remaining time is greater than 32767ms.
64+10	HR_VIBRATION_SET_SELECTED	R	1 to 26	This Holding Register gives the current selected vibration set.
64+11	HR_SEQUENCE_SELECTED	R	1 to 26	This Holding Register gives the current selected sequence.
64+12	HR_BACKLIGHT_STATE_STATUS	R	0: OFF 1: ON	This Holding Register gives the current backlight state
64+13	HR_READ_WRITE_PARAMETER_READ_VALUE	R	0 to 65535	The parameter read trigged with HR_READ_WRITE_PARAMETER_TRIG (Address 29) update this Holding Register value.
64+14	HR_FIRMWARE_VERSION_H	R	0 to 9	This Holding Register gives the first digit of the firmware version.
64+15	HR_FIRMWARE_VERSION_M	R	0 to 9	This Holding Register gives the second digit of the firmware version.
64+16	HR_FIRMWARE_VERSION_L	R	0 to 9	This Holding Register gives the third digit of the firmware version.
64+17	HR_USER_ACCESS_LEVEL	R	0: undefined	This Holding Register gives the current user access level. More information in

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
			1: operator 2: integrator 4: developer	Section 5.4.2.
64+18	HR_FLASH_OPERATIONS_STATUS	R	0: DONE 1: BUSY	This Holding Register gives the status of the flash operation trigged with HR_FLASH_OPERATIONS (Address 32).

Table 6-2: Status Holding Registers, offset=64

6.3.3. Recipe Read Holding Registers (Read-only zone, offset=128)

The Recipe Read Holding Registers presented in Table 6-3 are used to get the configuration values of the Asycube vibrations and sequences. They are directly linked to the values of the corresponding Asycube parameters (Chapter 4). Examples: frequency of the Actuator 2 for the vibration Forward 'A', duration of an action in the sequence 9.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
128+0	HR_PLATFORM_VIBRATION_A_AMPL1	R		Direct read access to corresponding parameter in Section 4.2.
128+1	HR_PLATFORM_VIBRATION_A_FREQ1	R		Direct read access to corresponding parameter in Section 4.2.
128+2	HR_PLATFORM_VIBRATION_A_PHASE1	R		Direct read access to corresponding parameter in Section 4.2.
128+3	HR_PLATFORM_VIBRATION_A_WAVEFORM1	R		Direct read access to corresponding parameter in Section 4.2.
128+4	HR_PLATFORM_VIBRATION_A_AMPL2	R		Direct read access to corresponding parameter in Section 4.2.
128+5	HR_PLATFORM_VIBRATION_A_FREQ2	R		Direct read access to corresponding parameter in Section 4.2.
128+6	HR_PLATFORM_VIBRATION_A_PHASE2	R		Direct read access to corresponding parameter in Section 4.2.
128+7	HR_PLATFORM_VIBRATION_A_WAVEFORM2	R		Direct read access to corresponding parameter in Section 4.2.
128+8	HR_PLATFORM_VIBRATION_A_AMPL3	R		Direct read access to corresponding parameter in Section 4.2.
128+9	HR_PLATFORM_VIBRATION_A_FREQ3	R		Direct read access to corresponding parameter in Section 4.2.
128+10	HR_PLATFORM_VIBRATION_A_PHASE3	R		Direct read access to corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
128+11	HR_PLATFORM_VIBRATION_A_WAVEFORM3	R		Direct read access to corresponding parameter in Section 4.2.
128+12	HR_PLATFORM_VIBRATION_A_AMPL4	R		Direct read access to corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
128+13	HR_PLATFORM_VIBRATION_A_FREQ4	R		Direct read access to corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
128+14	HR_PLATFORM_VIBRATION_A_PHASE4	R		Direct read access to corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
128+15	HR_PLATFORM_VIBRATION_A_WAVEFORM4	R		Direct read access to corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
128+16	HR_PLATFORM_VIBRATION_A_DURATION	R		Direct read access to corresponding parameter in Section 4.2.
128+17 to 128+441	HR_PLATFORM_VIBRATION_B_TO_Z_DATA	R		Direct read access to corresponding parameters in Section 4.2. The address has to be computed based on the 17 previous parameters (Address 128+0 to 128+16) with the following formula: Desired_Address = Address_Of_The_Register_For_Vibration_A + 17 x Vibration_Identifier, where 'A'=0, 'B'=1, … Example: HR_PLATFORM_VIBRATION_**E**_WAVEFORM3 = HR_PLATFORM_VIBRATION_A_WAVEFORM3 + 17 x **E** = 128+11 + 17 x **4** = 128+79
128+442	HR_HOPPER_A_DIGITAL_OUTPUT1	R		Direct read access to corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
128+443	HR_HOPPER_A_ANALOG_OUTPUT1	R		Direct read access to corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
128+444	HR_HOPPER_A_DIGITAL_OUTPUT2	R		Direct read access to corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
128+445	HR_HOPPER_A_ANALOG_OUTPUT2	R		Direct read access to corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
128+446	HR_HOPPER_VIBRATION_A_AMPL	R		Direct read access to corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
128+447	HR_HOPPER_VIBRATION_A_FREQ	R		Direct read access to corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
128+448	HR_HOPPER_VIBRATION_A_WAVEFORM	R		Direct read access to corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
128+449	HR_HOPPER_A_DURATION	R		Direct read access to corresponding parameter in Section 4.3.
128+450 to 128+649	HR_HOPPER_VIBRATION_B_TO_Z	R		Direct read access to corresponding parameters in Section 4.3. The address has to be computed based on the 8 previous parameters (Address 128+442 to 128+449) with the following formula: Desired_Address = Address_Of_The_Register_For_Vibration_A + 8 x Vibration_Identifier, where 'A'=0, 'B'=1, … Example: HR_HOPPER_**R**_ANALOG_OUTPUT1 = HR_HOPPER_A_ANALOG_OUTPUT1 + 8 x **R** = 128+443 + 8 x **17**= 128+579
128+650	HR_SEQUENCE_ID1_ACTION1_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+651	HR_SEQUENCE_ID1_ACTION1_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+652	HR_SEQUENCE_ID1_ACTION1_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+653	HR_SEQUENCE_ID1_ACTION1_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.
128+654	HR_SEQUENCE_ID1_ACTION2_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+655	HR_SEQUENCE_ID1_ACTION2_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+656	HR_SEQUENCE_ID1_ACTION2_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+657	HR_SEQUENCE_ID1_ACTION2_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
128+658	HR_SEQUENCE_ID1_ACTION3_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+659	HR_SEQUENCE_ID1_ACTION3_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+660	HR_SEQUENCE_ID1_ACTION3_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+661	HR_SEQUENCE_ID1_ACTION3_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.
128+662	HR_SEQUENCE_ID1_ACTION4_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+663	HR_SEQUENCE_ID1_ACTION4_VIBRATION	R		Direct read access to corresponding parameter in Section 5.4.10.
128+664	HR_SEQUENCE_ID1_ACTION4_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+665	HR_SEQUENCE_ID1_ACTION4_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.
128+666	HR_SEQUENCE_ID1_ACTION5_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+667	HR_SEQUENCE_ID1_ACTION5_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+668	HR_SEQUENCE_ID1_ACTION5_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+669	HR_SEQUENCE_ID1_ACTION5_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.
128+670	HR_SEQUENCE_ID1_ACTION6_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+671	HR_SEQUENCE_ID1_ACTION6_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+672	HR_SEQUENCE_ID1_ACTION6_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+673	HR_SEQUENCE_ID1_ACTION6_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.
128+674	HR_SEQUENCE_ID1_ACTION7_TYPE	R		Direct read access to corresponding parameter in Section 4.4.
128+675	HR_SEQUENCE_ID1_ACTION7_VIBRATION	R		Direct read access to corresponding parameter in Section 4.4.
128+676	HR_SEQUENCE_ID1_ACTION7_DURATION_MODE	R		Direct read access to corresponding parameter in Section 4.4.
128+677	HR_SEQUENCE_ID1_ACTION7_DURATION_VALUE	R		Direct read access to corresponding parameter in Section 4.4.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
128+678 to 128+1377	HR_SEQUENCE_ID2_TO_26	R		The address has to be computed based on the 28 previous parameters (Address 128+650 to 128+677) with the following formula: Desired_Address = Address_Of_The_Register_For_Sequence 1 + 28 x Sequence_Identifier Example: HR_SEQUENCE_ID26_ACTION7_DURATION_VALUE = HR_SEQUENCE_ID1_ACTION7_DURATION_VALUE + 28 x (**26**-1) = 128+677 + 28 x 25 = 128+1377

Table 6-3: Recipe Read Holding Registers, offset=128

6.3.4. Recipe Write Holding Registers (Write-only zone, offset=1536)

The Recipe Write Holding Registers presented in Table 6-4 are used to set the configuration values of the Asycube vibrations and sequences.

The Holding Registers highlighted in **bold** are triggers that start an action or change a value.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
1536+0	**HR_PLATFORM_PARAMETERS_WRITE_TRIG**	W	0 or 1	A rising edge trigs the write of the platform vibration Holding Registers HR_PLATFORM_VIBRATION_AMPL1 (Address 1536+2) to HR_PLATFORM_VIBRATION_DURATION0 (Address 1536+18) to the vibration parameters (Section 4.2) for the identifier HR_PLATFORM_PARAMETERS_WRITE_ID.
1536+1	HR_PLATFORM_PARAMETERS_WRITE_ID	W	0 to 25	The platform vibration write trigged with HR_PLATFORM_PARAMETERS_WRITE_TRIG uses this Holding Register to save the values at the given identifier, where 0 = 'A', 1 = 'B', …
1536+2	HR_PLATFORM_VIBRATION_AMPL1	W		This value is written to the corresponding parameter in Section 4.2.
1536+3	HR_PLATFORM_VIBRATION_FREQ1	W		This value is written to the corresponding parameter in Section 4.2.
1536+4	HR_PLATFORM_VIBRATION_PHASE1	W		This value is written to the corresponding parameter in Section 4.2.
1536+5	HR_PLATFORM_VIBRATION_WAVEFORM1	W		This value is written to the corresponding parameter in Section 4.2.
1536+6	HR_PLATFORM_VIBRATION_AMPL2	W		This value is written to the corresponding parameter in Section 4.2.
1536+7	HR_PLATFORM_VIBRATION_FREQ2	W		This value is written to the corresponding parameter in Section 4.2.
1536+8	HR_PLATFORM_VIBRATION_PHASE2	W		This value is written to the corresponding parameter in Section 4.2.
1536+9	HR_PLATFORM_VIBRATION_WAVEFORM2	W		This value is written to the corresponding parameter in Section 4.2.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
1536+10	HR_PLATFORM_VIBRATION_AMPL3	W		This value is written to the corresponding parameter in Section 4.2.
1536+11	HR_PLATFORM_VIBRATION_FREQ3	W		This value is written to the corresponding parameter in Section 4.2.
1536+12	HR_PLATFORM_VIBRATION_PHASE3	W		This value is written to the corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
1536+13	HR_PLATFORM_VIBRATION_WAVEFORM3	W		This value is written to the corresponding parameter in Section 4.2.
1536+14	HR_PLATFORM_VIBRATION_AMPL4	W		This value is written to the corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
1536+15	HR_PLATFORM_VIBRATION_FREQ4	W		This value is written to the corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
1536+16	HR_PLATFORM_VIBRATION_PHASE4	W		This value is written to the corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
1536+17	HR_PLATFORM_VIBRATION_WAVEFORM4	W		This value is written to the corresponding parameter in Section 4.2. N/A for the Asycube 50 and 80.
1536+18	HR_PLATFORM_VIBRATION_DURATION0	W		This value is written to the corresponding parameter in Section 4.2.
1536+19	**HR_HOPPER_PARAMETERS_WRITE_TRIG**	W	0 or 1	A rising edge trigs the write of the hopper vibration Holding Registers HR_HOPPER_DIGITAL_OUTPUT1 (Address 1536+21) to HR_HOPPER_DURATION0 (Address 1536+29) to the vibration parameters (Section 4.3) for the identifier HR_HOPPER_PARAMETERS_WRITE_ID.
1536+20	HR_HOPPER_PARAMETERS_WRITE_ID	W	0 to 25	The hopper vibration write trigged with HR_HOPPER_PARAMETERS_WRITE_TRIG uses this Holding Register to save the values at the given identifier, where 0 = ‘A’, 1 = ‘B’, …
1536+21	HR_HOPPER_DIGITAL_OUTPUT1	W		This value is written to the corresponding parameter in Section 4.3.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
				N/A for the Asycube 50 and 80.
1536+22	HR_HOPPER_ANALOG_OUTPUT1	W		This value is written to the corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
1536+23	HR_HOPPER_DIGITAL_OUTPUT2	W		This value is written to the corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
1536+24	HR_HOPPER_ANALOG_OUTPUT2	W		This value is written to the corresponding parameter in Section 4.3. N/A for the Asycube 50 and 80.
1536+25	HR_HOPPER_VIBRATION_AMPL	W		This value is written to the corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
1536+26	HR_HOPPER_VIBRATION_FREQ	W		This value is written to the corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
1536+27	HR_HOPPER_VIBRATION_WAVEFORM	W		This value is written to the corresponding parameter in Section 4.3. Only applicable for the Asycube 50 and 80.
1536+28	HR_HOPPER_DURATION0	W		This value is written to the corresponding parameter in Section 4.3.
1536+29	**HR_SEQUENCE_PARAMETERS_WRITE_TRIG**	W	0 or 1	A rising edge trigs the write of the hopper vibration Holding Registers HR_SEQUENCE_ACTION1_TYPE (Address 1536+31) to HR_SEQUENCE_ACTION7_DURATION_VALUE (Address 1536+58) to the sequence parameters (Section 4.4) for the identifier H HR_SEQUENCE_PARAMETERS_WRITE_ID
1536+30	HR_SEQUENCE_PARAMETERS_WRITE_ID	W	1 to 25	The sequence write trigged with HR_SEQUENCE_PARAMETERS_WRITE_TRIG uses this Holding Register to save the values at the given identifier. Note: The Sequence ID 26 is protected so read-only.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
1536+31	HR_SEQUENCE_ACTION1_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+32	HR_SEQUENCE_ACTION1_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+33	HR_SEQUENCE_ACTION1_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+34	HR_SEQUENCE_ACTION1_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.
1536+35	HR_SEQUENCE_ACTION2_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+36	HR_SEQUENCE_ACTION2_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+37	HR_SEQUENCE_ACTION2_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+38	HR_SEQUENCE_ACTION2_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.
1536+39	HR_SEQUENCE_ACTION3_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+40	HR_SEQUENCE_ACTION3_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+41	HR_SEQUENCE_ACTION3_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+42	HR_SEQUENCE_ACTION3_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.
1536+43	HR_SEQUENCE_ACTION4_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+44	HR_SEQUENCE_ACTION4_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+45	HR_SEQUENCE_ACTION4_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+46	HR_SEQUENCE_ACTION4_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.
1536+47	HR_SEQUENCE_ACTION5_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+48	HR_SEQUENCE_ACTION5_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+49	HR_SEQUENCE_ACTION5_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+50	HR_SEQUENCE_ACTION5_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.

Address (decimal)	Holding Register name	Read/ Write	Data Range	Comments
1536+51	HR_SEQUENCE_ACTION6_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+52	HR_SEQUENCE_ACTION6_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+53	HR_SEQUENCE_ACTION6_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+54	HR_SEQUENCE_ACTION6_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.
1536+55	HR_SEQUENCE_ACTION7_TYPE	W		This value is written to the corresponding parameter in Section 4.4.
1536+56	HR_SEQUENCE_ACTION7_VIBRATION	W		This value is written to the corresponding parameter in Section 4.4.
1536+57	HR_SEQUENCE_ACTION7_DURATION_MODE	W		This value is written to the corresponding parameter in Section 4.4.
1536+58	HR_SEQUENCE_ACTION7_DURATION_VALUE	W		This value is written to the corresponding parameter in Section 4.4.

Table 6-4: Recipe Write Holding Registers, offset=1536

6.4. Errors

Several types of error can arise in the Asycube used with Modbus TCP. The Holding Register HR_MODBUS_STATUS (Address 64) bits represent the status of these types of error.

Value (binary)	Name	Description
0b0001	Asycube Modbus errors status	If the value is 1, please refer to Section 6.4.2
0b0010	Internal Modbus communication error status	If the value is 1, please refer to Section 6.4.2
0b0100	Asycube warnings status	If the value is 1, please refer to Section 6.4.3
0b1000	Asycube alarms status	If the value is 1, please refer to Section 6.4.3

Table 6-5: Status register.

The error status can be reset by writing the corresponding bit to 1 in the Holding Register HR_MODBUS_CONTROL (Address 0). For example, writing the value 0b0110 (it corresponds to the value '6' in decimal) will reset both the Asycube Modbus errors status and the Asycube warnings status.

6.4.1. Main Modbus exception codes

The Modbus exception codes are defined in the protocol specification (http://www.modbus.org). A subset of these exceptions is implemented in the Asycube (See Table 6-6). The Modbus protocol is responsible to carry the exceptions directly in the Modbus message and are not set in a Holding Register.

Value (decimal)	Name	Description
1	Illegal Function	Function code received in the query is not recognized or allowed by slave
2	Illegal Data Address	Data address of some or all the required entities are not allowed or do not exist in slave
3	Illegal Data Value	Value is not accepted by slave

Table 6-6: Modbus protocol exception codes.

6.4.2. Holding Register: error codes

The HR_MODBUS_ERROR_CODE (Address 65) gives the current Asycube Modbus error.

The first byte (mask 0x00FF) gives the error linked to the Asycube Modbus implementation (See Table 6-7).

Value (hexadecimal)	Name	Description
0x01	Write Access Not Allowed	The data cannot be written because it is forbidden or the index is out of range. An example would be selecting a sequence (HR_SELECT_SEQUENCE) with a value out range such as 27.

Table 6-7: Asycube Modbus error codes.

The second byte (mask 0xFF00) gives the internal Modbus communication error code (See the value Error Bit in Table 5-2). The internal Modbus communication error typically occurs when a vibration is trigged with incorrect parameters values. An example would be trigging a platform centering (HR_PLATFORM_CENTERING_TRIG) while the value of the X position (HR_PLATFORM_CENTERING_X) is out of range, such as 120%.

6.4.3. Asycube warnings and alarms

The Asycube has its own warnings and alarms. They can be read respectively in the Holding Registers HR_WARNINGS (Address 66) and HR_ALARMS (Address 67). Their values are explained in the Table 5-20 and Table 5-21.

6.5. Example of use

This section presents some examples showing how to **control** the Asycube through Modbus TCP in production mode. We highly recommend using the Asycube HMI to **configure** the vibrations parameters as described in the Asycube User Manual.

6.5.1. Set up the Modbus master (e.g. PLC)

- Configure the Modbus master (=client) to communicate with the Asycube based on the Asycube communication parameters (Section 6.1). Don't forget to change your master network settings (IP address, …).
- Define a memory map of 16bits registers for data to be written in the Asycube. We recommend creating as many registers as contained in the Control Holding Registers (Section 0).
- Define a memory map of 16bits registers for data to be read from the Asycube. We recommend creating as many registers as contained in the Status Holding Registers (Section 0).
- Configure the Modbus master to alternatively write the Control Holding Registers and read the Status Holding Registers, with a cycle time of 10 ms.

6.5.2. Handle the Status and Control Holding Registers

- Program your Modbus master to monitor the Asycube errors state by reading the HR_MODBUS_STATUS register. Analyze the HR_MODBUS_ERROR_CODE, HR_WARNINGS and HR_ALARMS registers if needed.
- Program your Modbus master to be able to clear the errors by writing the HR_MODBUS_CONTROL register.

6.5.3. Vibration, sequence, backlight, read/write parameter

- Start a platform vibration
  - Set the vibration identifier in HR_PLATFORM_VIBRATION_ID, for example 3 for 'D', which corresponds to a 'Left' direction (according to the convention presented in Section 2.1).
  - Set the vibration duration in HR_PLATFORM_VIBRATION_DURATION, for example 1500 to prepare a vibration of 1.5 s.
  - Start the vibration by changing the value of the trigger HR_PLATFORM_VIBRATION_TRIG from 0 to 1.
- Monitor a platform vibration
  - Read the HR_MODBUS_PLATFORM_STATUS. If the value is ERROR, refer to the HR_MODBUS_STATUS for more information. A BUSY value means that the command has been understood and that the vibration has started.
  - Monitor the HR_MODBUS_PLATFORM_STATUS and wait for its value to get back to DONE, which means that the vibration is completed.
  - Or monitor the HR_PLATFORM_REMAINING_TIME to see the remaining vibration time. The value decreases to 0 when the vibration is completed.
- Start and monitor a sequence
  - Prepare the vibration parameters in HR_SEQUENCE_EXECUTION_NBPARTS, HR_SEQUENCE_EXECUTION_NBMAX, HR_SEQUENCE_EXECUTION_X, HR_SEQUENCE_EXECUTION_Y and HR_SEQUENCE_EXECUTION_SEQUENCEID.
  - Start the sequence by changing the value of the trigger HR_SEQUENCE_EXECUTION_TRIG from 0 to 1.
  - Monitor HR_MODBUS_SEQUENCE_STATUS and HR_SEQUENCE_REMAINING_TIME and act accordingly.
- Stop a sequence currently running
  - Stop the sequence by changing the value of the trigger HR_SEQUENCE_HALT from 0 to 1.
- Switch the backlight ON

  - Change the value of HR_BACKLIGHT_STATE_CONTROL from 0 to 1.
- Read a parameter value
  - Prepare the read action by setting the parameter address in HR_READ_WRITE_PARAMETER_ADDRESS, according to Chapter 4 (make sure to have an even address).
  - Start reading by changing the value of the trigger HR_READ_WRITE_PARAMETER_TRIG from 0 to 1.
  - Read and use the result value in HR_READ_WRITE_PARAMETER_READ_VALUE.
- Write a parameter value
  - Prepare the write action by setting the parameter address in HR_READ_WRITE_PARAMETER_ADDRESS+1, according to Chapter 4 (make sure to have an odd address).
  - Prepare the value to write in HR_READ_WRITE_PARAMETER_WRITE_VALUE.
  - Start writing by changing the value of the trigger HR_READ_WRITE_PARAMETER_TRIG from 0 to 1.