



ITalks MCS 1608

Technical Product Description Digital Inputs

General description

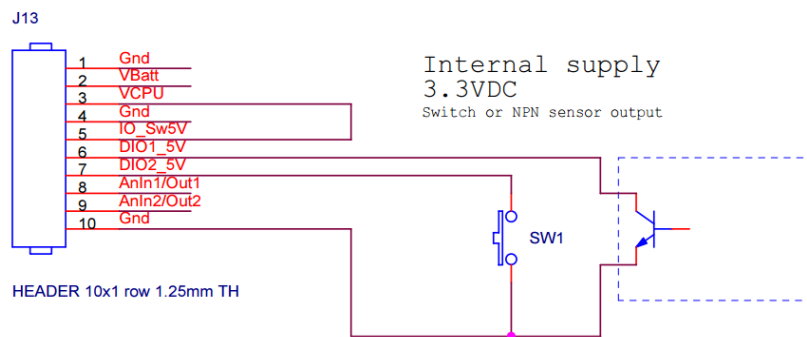
This document is intended to explain the use of digital inputs on the MCS 1608 Full and I/O version. The digital inputs (#2) can be used at different modes. Configuring them is a little bit harder because the digital inputs disable the RX and TX lines of the UART, used for the terminal/debug tool. In order to configure them a good downlink channel is needed!

Pre requisites

- An MCS 1608 Full version with Digital Input firmware enabled (V2.2 or higher)
- A monitor/debug cable (ask your supplier of the MCS 1608)
- A digital input cable (ask your supplier of the MCS 1608)
- A terminal/debug tool
- A good understanding of the MCS 1608, how it works and how to send commands using downlink messages

Good to know

The MCS 1608 has 2 digital inputs connected to the 10Pin connector on the PCB. The internal circuitry for digital inputs of the MCS 1608 must be powered externally. Using the original MCS 1608 digital input cable (available as an accessory) takes care of this:

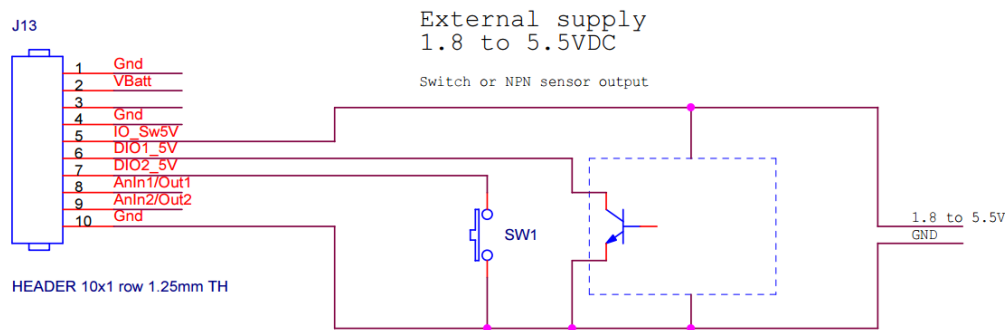




ITalks MCS 1608

Technical Product Description Digital Inputs

It is also possible to power the internal circuitry yourself:



Pin6 is connected to DI1 this pin is also used for the RX of the terminal. Enabling DI1 will disable receiving anything from the terminal/debug tool. On the original cable this is the RED-wire

Pin7 is connected to DI2 this pin is also used for the TX of the terminal. Enabling DI2 will disable sending anything to the terminal/debug tool. On the original cable this is the YELLOW-wire

It is therefore recommended if you only use one digital input to use DI1 because you can still use the monitor debug tool to see what is going on. It is necessary to have an active LoRa or SigFox connection with downlink capability to configure the digital inputs.

!If you know your configuration and you want to roll-out large numbers of MCS 1608, please contact MCS. We can provide the hardware in larger volumes with your custom firmware with default setting you need for your application!

!During SigFox transmission the Digital inputs are not processed!

!The digital inputs have pull-up resistor. They use power when they are forced to zero. Use a normally open contact!

Digital Inputs/Counters

The digital inputs are used as a counter and they can register running hours. As long as the Digital Input is low, the running hour timer is increased every second.



ITalks MCS 1608

Technical Product Description Digital Inputs

A digital input can be set in different modes:

- 0 = Off: The digital inputs is not used (serial interface can be used). This is the default Mode.
- 1 = Switch: The digital input can be connected to a (mechanical) switch. Debouncing is enabled (5 ms) to prevent one switch activation to lead to a multiple increase of the counter due to a bouncing contact. The maximum frequency is therefore < 200Hz. The counter will increase on every negative edge of the digital input.
This is the recommended setting for most use cases!
- 2 = Counter: The digital input can be connected to a fast digital signal. It will increase a counter on each negative edge of the input. The counter signal can be 10KHz. The counter will increase on every negative edge of the digital input.
!the Alarm report mode will not work when DI is configured as a counter!
- 3 = SlowSwitch: The MCS 1608 will monitor the digital input, when the digital input changes (debouncing enabled) it will wait for the signal to stabilize during a hold on/off time. When the signal is still in the same (changed) position when the hold on/off time finishes the counter is increased. The counter will increase on every edge (negative and positive) of the digital input.

The digital input can (currently) report in three different ways:

- 1 = Alarm, an alarm message is send on an increase in the counter of the digital input.
!This reporting mode cannot be used when the DI is configured as a counter!
- 2 = Time based, on a regular interval (only when the counter has changed states) a messages is send containing the counter value and the running time
- 3 = Count based, after XX counts a message is send containing the counter value and the running time

The Digital input can be configured using three parameters:

- Mode, see above value 0..3 (Off, Switch, Counter, Slowswitch)
- ReportMode, see above value 1..3 (Alarm, Time Based, Count Based)
- Hold On/Off time, the delay in processing the digital input is SlowSwitch mode, value 1.255 (1 = 10 sec, 2 = 20 sec etc.)
- Report Value, the interpretation of this value depends on the reporting mode
 - Alarm, the value is not used
 - Time Based, the value is the number of minutes between transmissions of the DigIn message



ITalks MCS 1608

Technical Product Description Digital Inputs

- Count Based, the value is the number of counts between transmissions of the DigIn message

Parameters

The following parameters can be changed via downlink messages

Parameter	Description	Factory Value	Command to change
#DigIn1Mode	0 = Off 1 = Switch 2 = Counter 3 = SlowSwitch (see description)	0 (Not Used)	0x21
#DigIn1ReportMode	1 = Alarm 2 = Time based 3 = Count based (see description)	3 (Count Based)	0x22
#DigIn1Hold On/off Time	1..255 (see description)	1 (10 Sec)	0x23
#DigIn1ReportValue	1..255 (see description)	10 (10 minutes) 10 (10 Counts)	0x24
#DigIn2Mode	0 = Off 1 = Switch 2 = Counter 3 = SlowSwitch (see description)	0 (Not Used)	0x25
#DigIn2ReportMode	1 = Alarm 2 = Time based 3 = Count based (see description)	3 (Count Based)	0x26
#DigIn2Hold On/off Time	1..255 (see description)	1 (10 Sec)	0x27
#DigIn2ReportValue	1..255 (see description)	10 (10 minutes) 10 (10 Counts)	0x28

Commands

Commands are defined by their number as mentioned in the table above.
Commands are sent as downlink messages.



ITalks MCS 1608

Technical Product Description Digital Inputs

The downlink message format is as follows:

```
typedef struct {  
    byte CmdSeq;           // Should be incremented every command given  
    byte Cmd;             // Command number as mentioned in table above  
    byte NewValue;       // The new value of the parameter to be changed  
} TGenSensCmd;
```

A command is answered by the MCS 1608 via a [TAliveMsg](#). The parameter CmdAck in this message contains the last received command.

There are a number of special commands:

Reboot command

has the following content:

```
byte CmdSeq = 0xFE  
byte Cmd = 0xFE  
byte NewValue = 0xEF
```

the MCS 1608 will perform a reboot

Factory Reset command

has the following content:

```
byte CmdSeq = 0xEF  
byte Cmd = 0xFF  
byte NewValue = 0xFE
```

The MCS 1608 will restore parameters to factory default and perform a reboot

Formal Payload description

General Sensor Message Format for Digital Inputs

This is the latest message format, to fully use the General Sensor capabilities.

There are currently 14 message types

- [MsgIDAlarm](#) [MsgID 0x04](#)
- [MsgIDDigIn1Msg](#) [MsgID 0x0C](#)
- [MsgIDDigIn2Msg](#) [MsgID 0x0D](#)

The messages always contain the most recent data. If no updated data is available the old data

is sent. The payloads can be decrypted via the 1M2M Payload decoding JSON service

<http://1m2m.eu/services/GETPAYLOAD?Human=0&PL=0102096100064f7a3c07a50300000000>

```
typedef struct {
```



ITalks MCS 1608

Technical Product Description Digital Inputs

```

byte MsgID; // Message Identification Value = 0x04
unsigned int GravRotAl :1; // Gravity Rotation Detected
unsigned int MagRot :1; // Magnetic Rotation Detected
unsigned int MotAlarm :1; // Motion Alarm detected
unsigned int GeoFenceAl:1; // GeoFence Violation Detected
unsigned int VibrAl :1; // Vibration Alarm Detected
unsigned int ShockAl :1; // Shock alarm (future use)
unsigned int DigIn1Al :1; // DigIn1 Alarm Detected
unsigned int DigIn2Al :1; // DigIn2 Alarm Detected

int16 Temp; // Temperature in 0,01 Celcius
byte Hum; // Relative Humidity in %
word BaromBar; // Air Pressure in Mbar=MsgIDMsgIDBaromBar
+100.000)/100) }TAlarmMsg;

typedef struct {
    byte MsgId; // Message Identification Value = 0x0C
    byte Mode; // Current Digin1 Mode
    byte RepMode; // Current Digin Report Mode
    dword Counter; // Digin1 Counter
    dword RunTimer; // Digin1 RunTimer (seconds)
    byte State; // State of Digin1
}TDigin1Msg;

typedef struct {
    byte MsgId; // Message Identification Value = 0x0D
    byte Mode; // Current Digin2 Mode
    byte RepMode; // Current Digin Report Mode
    dword Counter; // Digin2 Counter
    dword RunTimer; // Digin2 RunTimer (seconds)
    byte State; // State of Digin2
}TDigin2Msg;

```

Version History

V1.0	Baseline	24-10-2016	Initial Version
V1.2		03-01-2017	Minor Changes in Message format (state added)