Manual ARMMI version 3

Advanced Railroad Machine-to-Machine Interface

revision 1 - 2022-03-05

This document describes a product from Advanced Rail Systems (ARS) LLC. You may contact the company at the following:

+1 (254) 252-3005 (sales)

+1 (254) 252-3005 ext. 106(support)

www.advancedrailsystems.com

sales@advancedrailsystems.com

Headquarter at 418 Hwy 164 Riesel, Texas 76682

Revisions of this document

Rev	Author	Date	Notes
1	Gustavo Vargas	2022-06-05	
2	Michael Minor	2025-30-06	Updated I/O isolation bullet point

Table of Contents

1 Abbreviations

AIMCAN	ARMMI Inter-Module CAN-bus
AMP	ARMMI Management Protocol
ARMMI	Advanced Railroad Machine-to-Machine Interface
ASP	ARMMI Secure Protocol
bps	Bits Per Second
CAN-Bus	Controller Area Network Bus
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DTMF	Dual T1 Mult-Frequency
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
OLED	Organic Light Emitting Diode
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
TCP	Transfer Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus

2 Glossary

Genisys..........................Simple protocol normally used to control railroad devices though a serial channel.

Modbus .... An industrial protocol that is normally used to control remote devices. The serial variation is called Modbus/RTU and the Ethernet version is Modbus/TCP.

3 Introduction

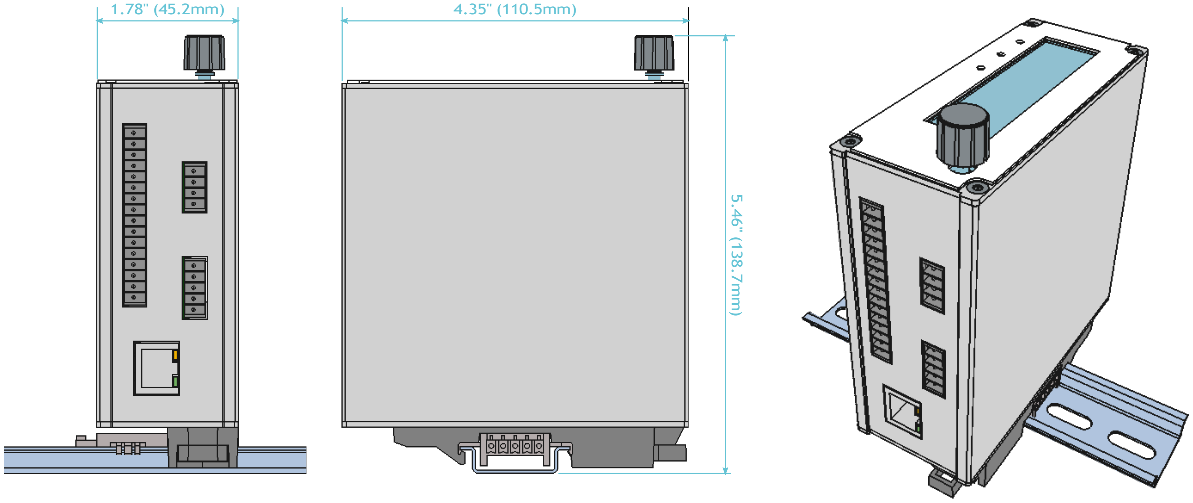
ARMMI is a programmable processor capable of controlling any yard switch machines or devices like single switch, crossovers, gates, bridges, crossings, derails and blue flag systems etc.

The ARMMI can also be programmed to execute any task compatible with its inputs/outputs, communication interfaces and processing capabilities, as described ahead in this manual.

The operation mode is similar to industrial PLC applications where all inputs and outputs are generic and their specific functions are defined by the currently installed program.

ARS offers a variety of pretested configurations for the more common applications and can also create a new applications for specific customer needs. Documentation of how to create your own configuration is available. Contact ARS's support for more information.

4 General Information



Part Number # 851-004 Features:

- Aluminum housing (4.35" x 5.46" x 1.78") (110mm x 139mm x 45mm)
- Operational voltage: 9 – 26Vdc

- Operational temperature: -40ºC to 85ºC
- Operation current: up to 60mA (with display off, 110mA when display is active)
- Core ARM Cortex-M4
- User Interface:
    - OLED Graphic Display with dimensions: 3/4” x 2-3/4” (20 x 71mm). The display can be used to show any sort of information:
        - Firmware and Project information and versions
        - Configuration parameters and communication addresses
        - Active inputs and outputs for diagnostic
        - Battery voltage
        - Temperature
        - Real time clock
        - Communication status
        - Any internal counter, flag, alarm, etc defined by the active Project
        - Automatic display off after 10 minutes of inactivity of push-button control
        - Any graphics added to the project
    - Encoder/Push-button for menu navigation
- Fail safety operation rated
- Inputs and Outputs
    - 16 outputs up to 3A each at the same voltage as the supply voltage
    - 8 of the outputs with load monitoring capabilities
    - all 16 outputs include a short-circuit and overload circuit that turns the output off and automatically resumes normal operation once the issue is fixed
    - 16 inputs that accept between 9Vdc and 26Vdc
    - 7 of 16 inputs and 4 of 16 outputs can be configured to use OSSD signal
    - 2 out of 16 outputs and 2 out of 16 inputs are of negatives type to facilitate interconnection with vital circuits (see page 12)
    - 2.500V optical isolation plus transient suppression in each I/O

- Communication interfaces (all terminals for the ports include surge protection up to 3.5kV)
  - USB mini-B for programming and configuration
  - Serial ports RS-232 for communication (spread spectrum, GPRS/GSM)
  - Serial RS-485C
  - 1 Ethernet 10/100 IPv4 communication port
  - 2 CANbus communication ports for remote operation
  - Input for remote control using handheld VHF radios with DTMF and voice messages (up to 32 messages with 20 seconds each)
- Embedded microphone to record voice messages
- Protocols
  - Genisys
  - Modbus TCP and RTU
  - ASP
  - MQTT
- Memory
  - Event log up to 72,000 events
  - Removable micro-SD card with all controller configuration for backup

4.1 Connectors

The ARMMI has a set of connectors on top and bottom of the housing to facilitate installation when using DIN-rail installation.

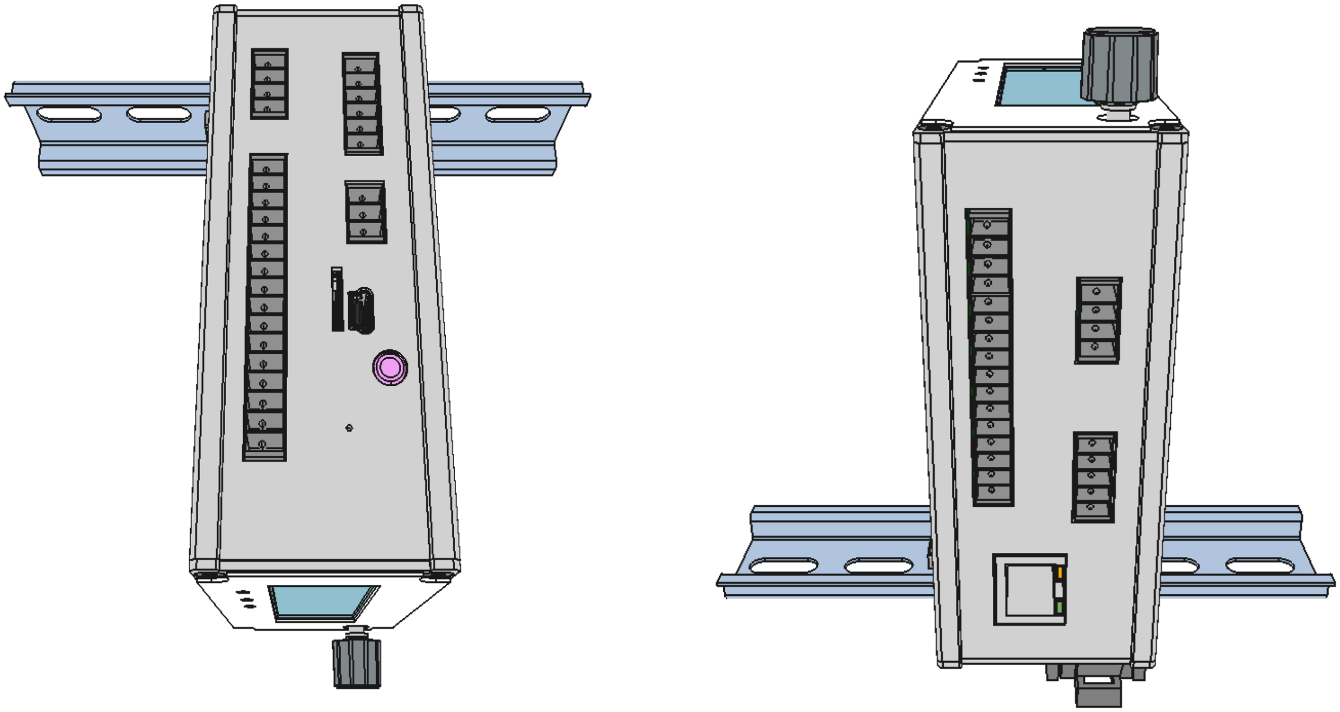
Most connector are intentionally designed in different sizes to ensure they are not connected at the wrong position when replacing processors in the field.

List of connectors and part-number reference:

Connector	Details	Part Number
Power	4 pin Wago connector	Wago - 714-104
Inputs	16 pin Wago connector	Wago - 714-116
Outputs	16 pin Wago connector	Wago - 714-116
RS-232	6 pin Wago connector	Wago - 714-106

RS-485	3 pin Wago connector	Wago - 714-103
Radio	4 pin Wago connector	Wago - 714-104
Ethernet	8P8C	Standard RJ45
Microph1	Plug P2 3.5mm mono	Standard P2 Connector

As mentioned during the introduction, Inputs and Outputs are not fixed to any special function, the installed application will define a function for each I/O and also define the protocol that will be active on the communication ports.



4.1.1 Power

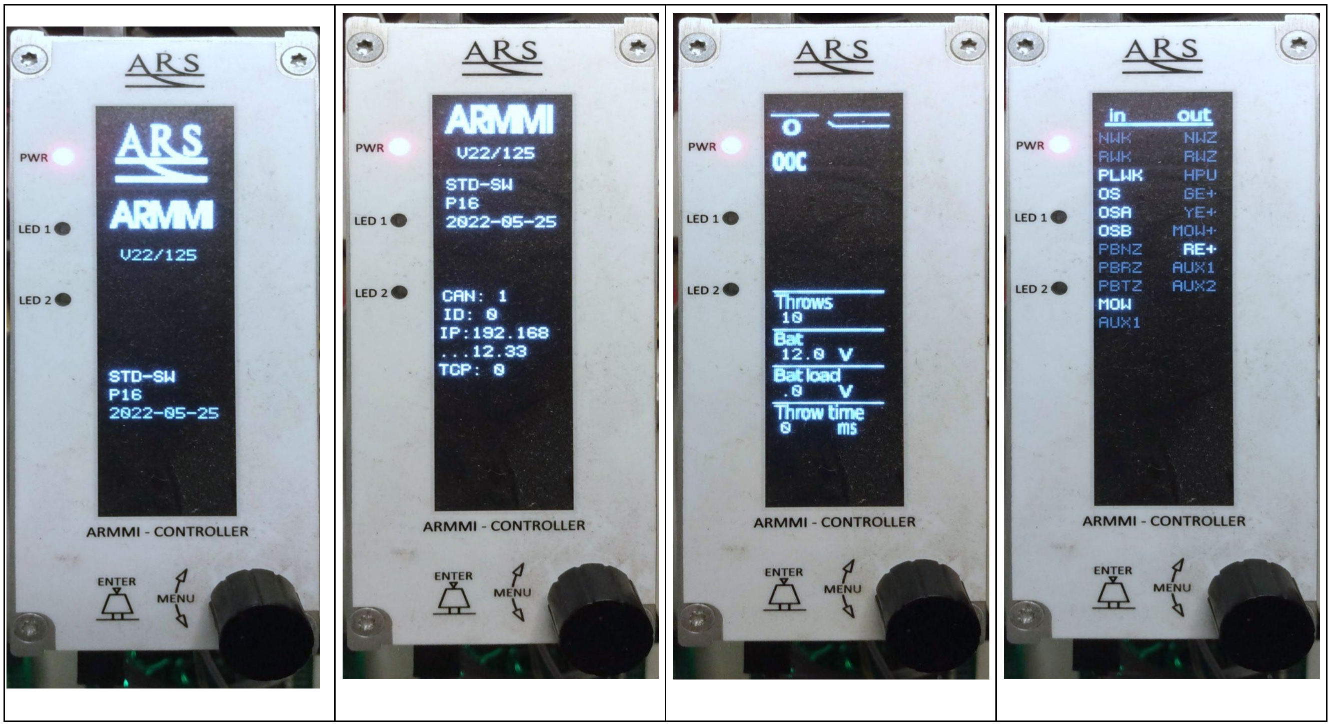
The power connector has connections for both NBAT and +BAT and are both internally connected. This is to facilitate daisy-chain connection to other modules on the DIN-rail. Any of the NBAT will work for the battery negative as any of the +BAT will work for the positive voltage. Any voltage between 9 and 26Vdc will work. User must keep in mind that the output voltage present on the Output connector pins will have the same level, so ensure loads (valves, light, etc.) are rated for this same voltage.

5 User Interface

The ARMMI includes an OLED display that is easily read during the day and night with a simple turn-and-click knob that is easy to use even with gloves.

The display will briefly show important information just after power-on. This information will be visible for about 3 seconds and then it will start showing the information defined by the installed configuration. The initial information includse:

- Version of installed firmware
- Name and version of installed project
- ID of communication interfaces



Above are example: First is the boot view with some project information. Seconds 1 is a standard view showing project name (STD-SW), version (P16), date of creation, and some communication parameters. Active CAN ID 1, unit ID (0), ethernet IP (192.168.12.33) and ethernet protocol port (0 in this example).

Views 3 and View 4 are specific application project file. Each project can be tailored to adjust the view as necessary to describe its current status. Project view panels may vary.

Example:

- a switch machine controller will show information about switch position, alarms, inputs status, communication link status, etc
- a z1 controller may show number of wheels in each z1, speed of movement, occupancy flags, etc

User can use the encoder to navigate between panels and access information.

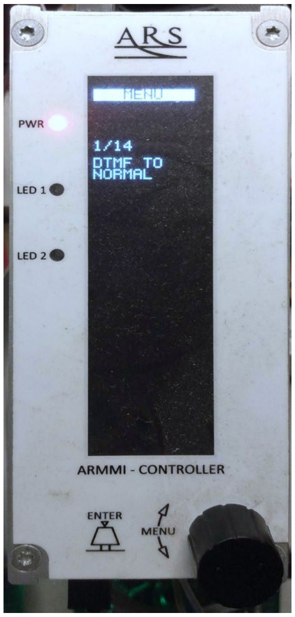
After 10 minutes of inactivity the display is set to sleep mode and turned off automatically. Click the encoder button to activate the display.

In most applications standard configuration items can be adjusted in the field thought the user-interface (instead of using the web interface describe in page 19).

To enter the configuration mode press and hold the encoder until the display changes to configuration mode and the MENU ribbon is at the top of the display.

User can use the encoder to navigate between panels and access information.

The number and types of configurations are defined by the installed Project file.

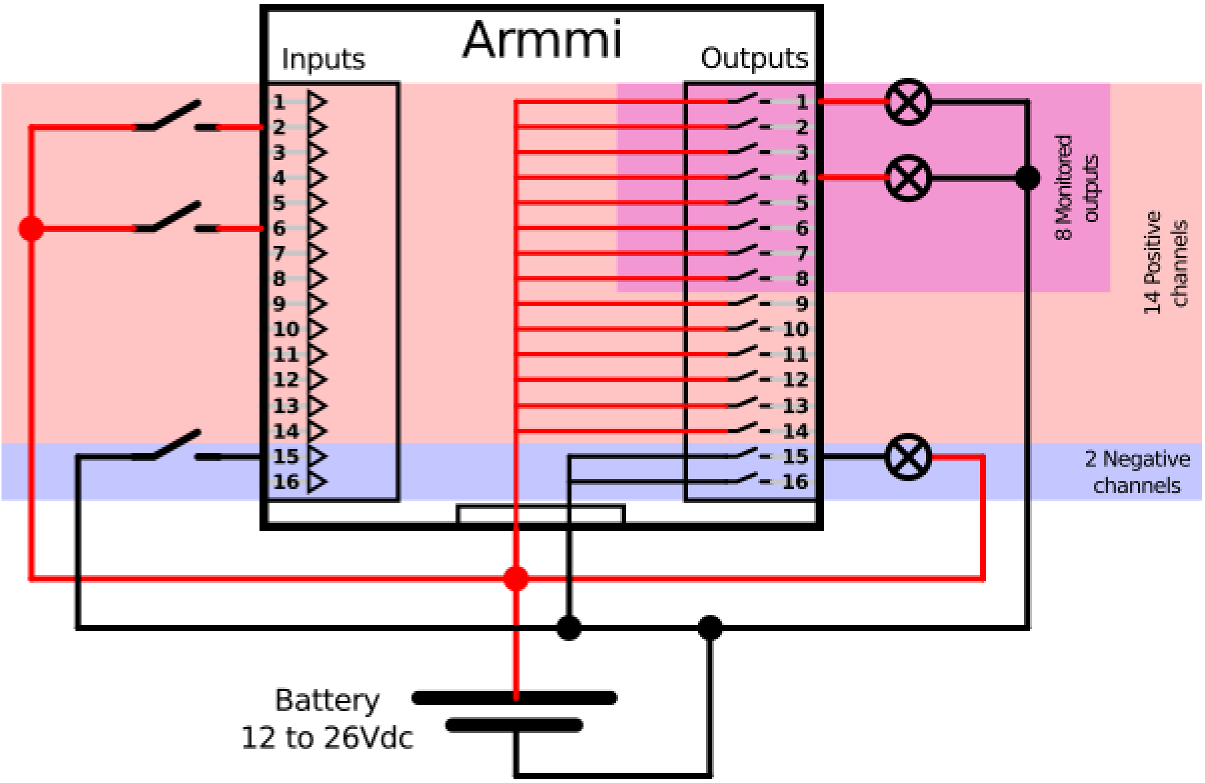


To Exit the configuration menuscroll to the last option in the menu and select “return” to resume normal operation. Typically, this “*Return*” option of the configuration menu will manage this automatically and restart the processor when necessary. Older configurations take place immediately after adjusting them, others will require the ARMMI processor to be restarted to implement the changes.

6 Inputs and Outputs

ARMMI processor includes up to 16 outputs and 16 inputs to control external load and read external commands/sensors.

The following diagram depicts the architecture of the inputs and outputs:



Positive inputs will accept any voltage ranging between 9Vdc up to 26Vdc to be considered active.

Negative inputs are active when connected to 0Vdc (battery's negative usually).

Positive outputs, when active can drive up to 3A with the same voltage from the external supply voltage. The first 8 positive outputs include circuitry that allows the ARMMI to confirm the presence of a load when the output is active.

Negative outputs are connected to 0Vdc (the external negative power supply), when active and can sink up to 2A each.

All 16 outputs include an overload protection that will shut the output down in case of overload or short-circuit. The program running on the processor will be informed of this failure only for the first 8 outputs. Output resumes normal operation automatically once the short-circuit is solved.

7 Communication Interfaces

7.1 USB 2.0 mini-B

The USB port is located on the top side of the processor. Its main usage is to upgrade firmware code, install a new application, configure the application and download current event log.

User must be running the ACS software on their computer to have access to these functions.

Please refer to chapter 8 on page **Error! Bookmark not defined.** for an overview of the software.

7.2 Serial Ports

ARMMI includes 4 serial ports being two CAN buses, 1 RS-232 and 1 RS-485C. While RS-232 can deal with different protocols as instructed by the Project the other 2 interfaces run on specific protocols. Check following description for more details.

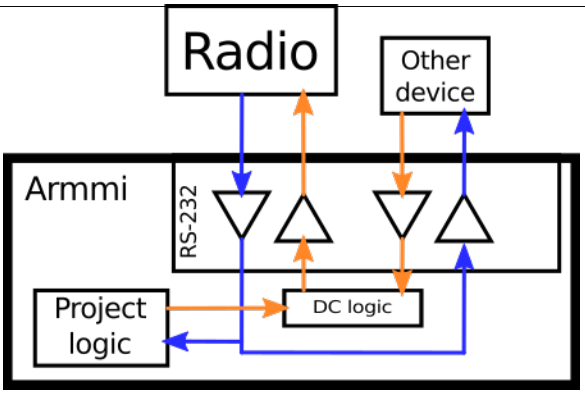
7.2.1 RS-232

This port can be used to communicate with any source of RS-232 signal. Main use cases are standard RS-232 ports, USB to RS-232 adapters, data radios, GPRS radios, etc.

The port can communicate with following parameters:

- 7 and 8 bits
- n1, odd and even parity
- 1 stop bit
- from 300 bps up to 250 kbps

This hardware includes a feature that allows for the RS-232 to be daisy-chained as follow:



This kind of architecture is protocol dependent.

With this setup we may connect a single data radio (or any other data source), to ARMMI’s input and any other device (as long it talks the exact same protocol in use by the ARMMI), to the daisy-chain output. **ARMMI and the other device must have different network IDs.** This setup allows for a single radio to communicate with 2 devices.

Internally the Project file will define this port to use 1 of Genisys, Modbus RTU, or Serial ASP.

ASP is NOT a standard protocol. It stands for ARMMIs’ Secure Protocol and its specification balances between the need to fulfill customers request for a secure channel and the power of ARMMI’s core. ASP uses UDP or serial link to communicate.

Both Genisys and Modbus included functions follows the official specifications.

Genisys functions:

- 0xF1 – requested data response
- 0xF2 – acknowledged acknowledge
- 0xFA – acknowledge received data
- 0xFB – Request modified index
- 0xFC – write index
- 0xFD – request full index

Modbus RTU functions:

- 0x03 – read multiple holding registers

- 0x04 – read multiple input registers
- 0x10 – write multiple holding registers

Please contact ARS to request more information about ASP protocol.

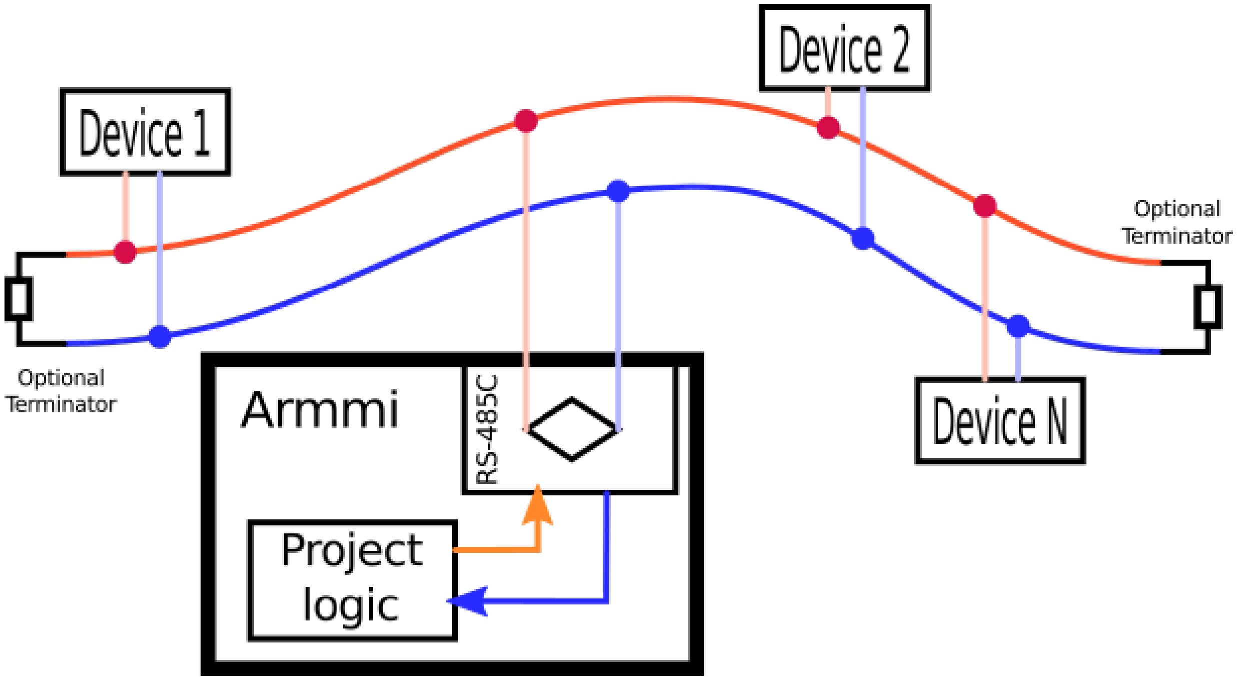
7.2.2 RS-485C

Available RS-485C port can operate using following parameters:

- 7 and 8 bits
- no parity
- 1 stop bit
- from 300 bps up to 130 kbps

The network topology must use a bus with very short leads (the line that connect the device to the bus). Ideally the bus should be a twisted pair with 120 ohm of characteristic impedance. When installed properly the bus terminators are going to increase the stability of the system, however they are not strictly necessary when running at low baud rates (this is our use case). ARMMI

internals will apply a bias and an extremely high termination to the bus line. Consult ARS for details and proper specification of a suitable external terminator if necessary.



If the bus cable includes a shield, it can be connected to the reference signal provided at the RS-485C connector, again this is not mandatory for this bus.

Current firmware version uses this port exclusively to communicate with wheel counters. This allows for up to 21 external wheel counters sensors to be connected to a single ARMMI and the logic to create the protection zones is handled by the installed application code. This also allows sharing the current counting of each zone with the other communication interfaces for remote monitoring.

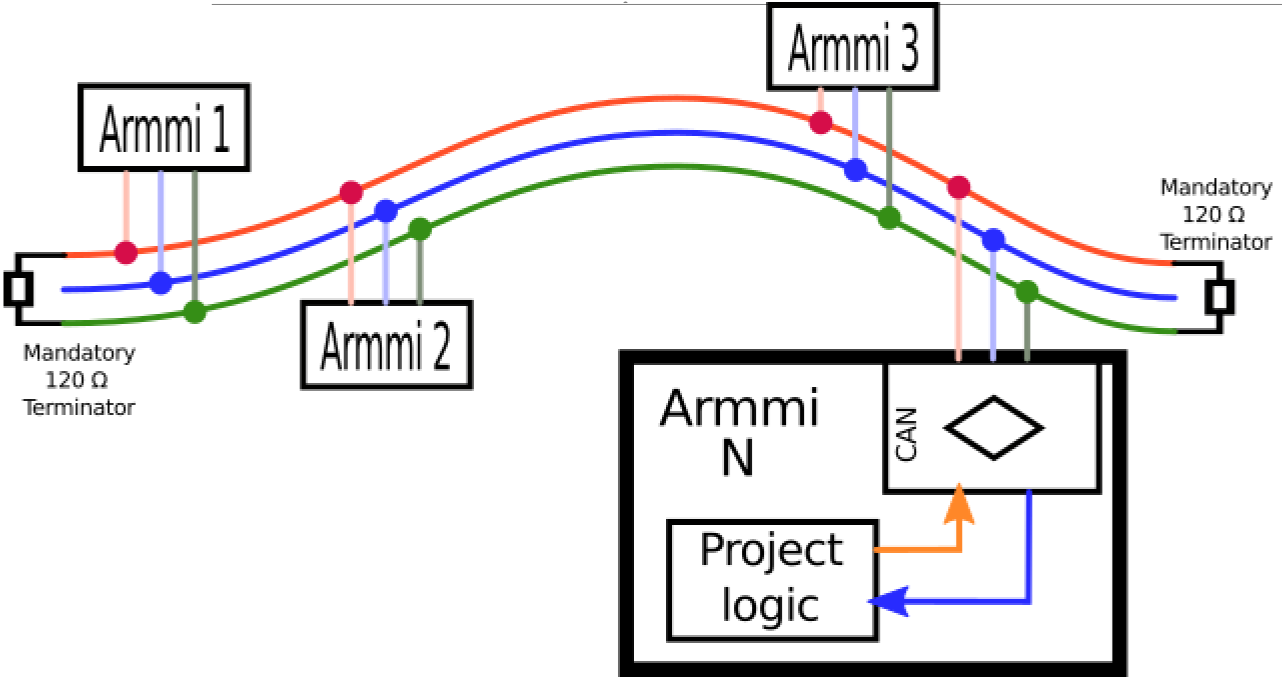
7.2.3 CAN Bus

ARMMI includes 2 CAN interfaces. They are designed specifically to interconnect multiple ARMMI in a closed network. High level protocol was custom designed for this usage to not incentive the idea of connecting others CANBus devices to the ARMMI network. This would be a desire if using standard CANopen or J1939 protocols for example. When multiple ARMMIs are connected together they start sharing internal flags, register and signals and this allow for the interaction between Projects to control each other.

Crossovers are a simple use case where 2 units are connected to allow two machines to be controlled in sync from a single point.

The topology is very similar with the 1 used by RS-485, with 3 important differences:

- Terminators are mandatory
- Shield connection is mandatory
- Stubs should be as short as possible



Each ARMMI includes an internal terminator that must be activate by the installation of a jump at the CAN-bus connector. So, in the example above, user could install the jump on units 1 and N since they are at the ends of the bus cable. Or use an external terminator as showed at the figure. **Just be sure to have exactly two terminators and that they are at the ends of the bus cable**.

A proper terminated bus will measure 60 ohm of resistance between its wires.

7.3 Ethernet Port

1 standard Ethernet 10/100 IPv4 port is available. When active, their parameters can be adjusted by the ACS software or by the web interface.

For security reasons this port will not work with DHCP or any other automatic configuration protocol.

User must manually set the following parameter for the port to properly work:

- unit IP
- MAC (optional since processor is shipped with a unique Local/Unicast MAC code)
- Sub-net mask
- Gateway IP
- DNS IP (currently not in use since this device will operate as a server exclusively)

- SNTP IP (optional)

User also may set following parameters:

- Enable internal HTTP server for remote configuration
- Set TCP/UDP port for modbus/TCP protocol
- Set UDP port for Genisys protocol
- Set UDP port and keys for ASP protocol

Modbus/TCP implements the same functions described in Modbus/RTU on chapter 7.2.1. Refer to same chapter to check out functions used by Genisys UDP.

When considering the protocol ports, the user must keep in mind that, even though it is possible to set all intended ports, only one protocol will be active for each application. Check applications specifications to see the protocol being used.

HTTP protocol will be active at the same time on port standard 80 and allows for users to connect to the ARMMI using a standard web browser to configure it and check the event log.

Note it is a HTTP server and not a secure HTTPS server. User is responsible to check with their internal IT department to discuss about security concerns related to this subject.

7.4 DTMF radio interface

The ARMMI includes an interface to a VHF radio that allow users to use a standard handheld radio to send DTMF commands to the processor and receive the pre-recorded response messages that are triggered by the running application.

This ability allows the safe operation of equipment outside the operators view with multiple commands and receive audible messages to confirm the command was properly executed or the reason it failed.

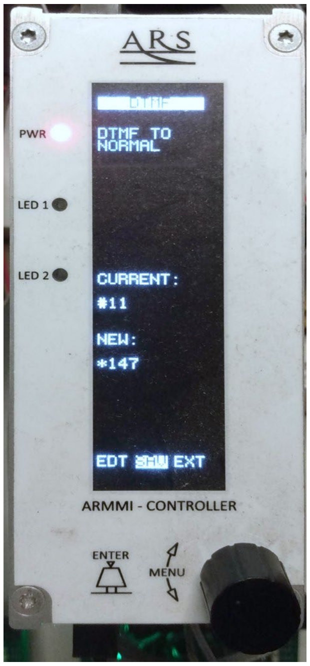
Applications can define up to 32 DTMF commands and playback up to 32 voice messages 20 seconds in length

User can change the DTMF codes using the web interface (check page **Error! Bookmark not defined.** for that), or though the push button user interface.

To execute the change using the user interface:

1. Enter the configuration menu (hold button until it activates)
2. walk to the DTMF code that need to be changed

3. click to activate it
4. type the new DTMF on your handheld radio
5. Check the screen if the code is correct. If not simple pause for 2 seconds and type again
6. If new code is correct, click to activate the operation menu and move to the Desired option:
    1. **EDT** (Edit) to resume editing the code
    2. **SAV** (Save) to save the new code and leave
    3. **EXT** (Exit) to leave without saving
7. Walk to the last menu item "*Return" and click to leave the configuration menu and resume normal operation*



DTMF

DTMF TO NORMAL

CURRENT:
#11

NEW:
*147

EDT SAV EXT

Unlike DTMF, voice message can only be recorded using the web interface. Check page 22.

8 Configuration Interface

Local user interface allows user to set simpler parameters directly in ARMMI’s screen. For more specific settings, project/firmware upgrade and event log access user must use the embedded web interface.

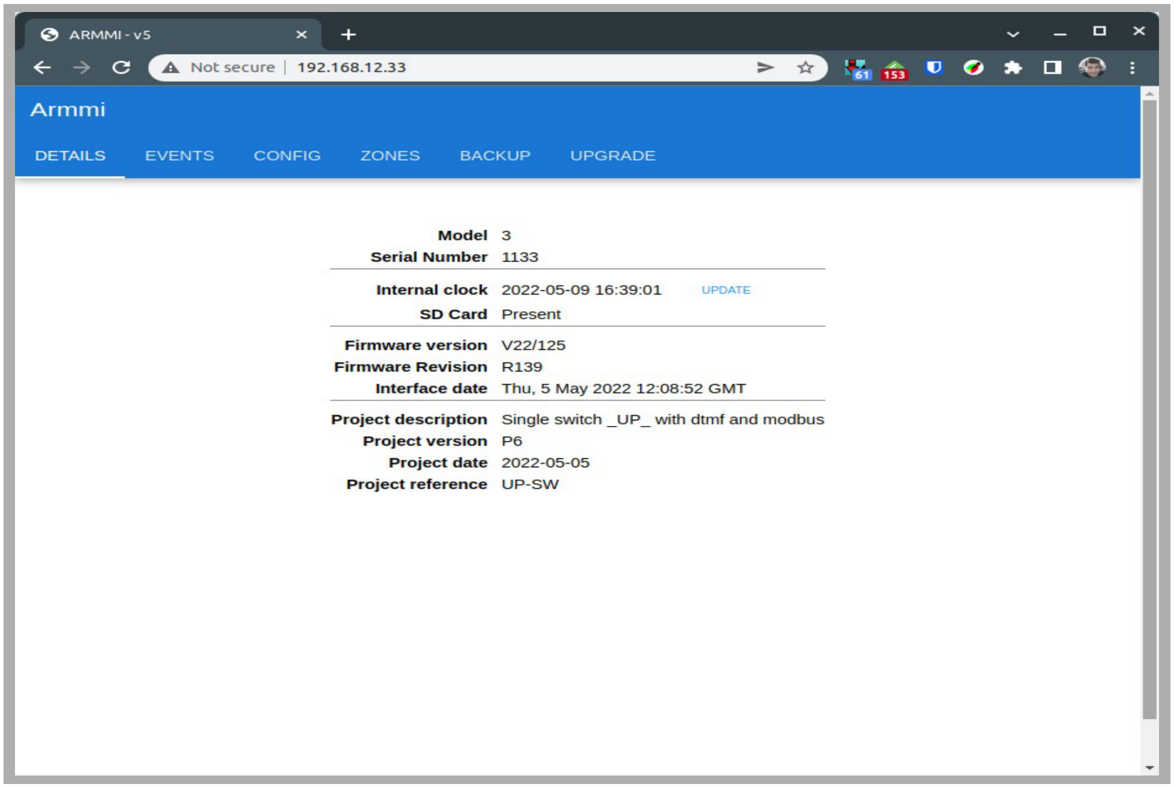
It is always available at standard HTTP port 80 and can be accessed by entering current ARMMI’s IP on any standard web browser (check current IP in ARMMI’s screen if necessary, default IP is 192.168.12.33)

It is possible to access it through a short cable between a laptop and ARMMI or remotely using the customer network if ARMMI is connected to it.

Once connected to the interface, user must enter unit password (defaults to 123123), before having access to unit’s data.

8.1 Web Interface

The first page of the web interface details information about current unit installed firmware and installed project for reference



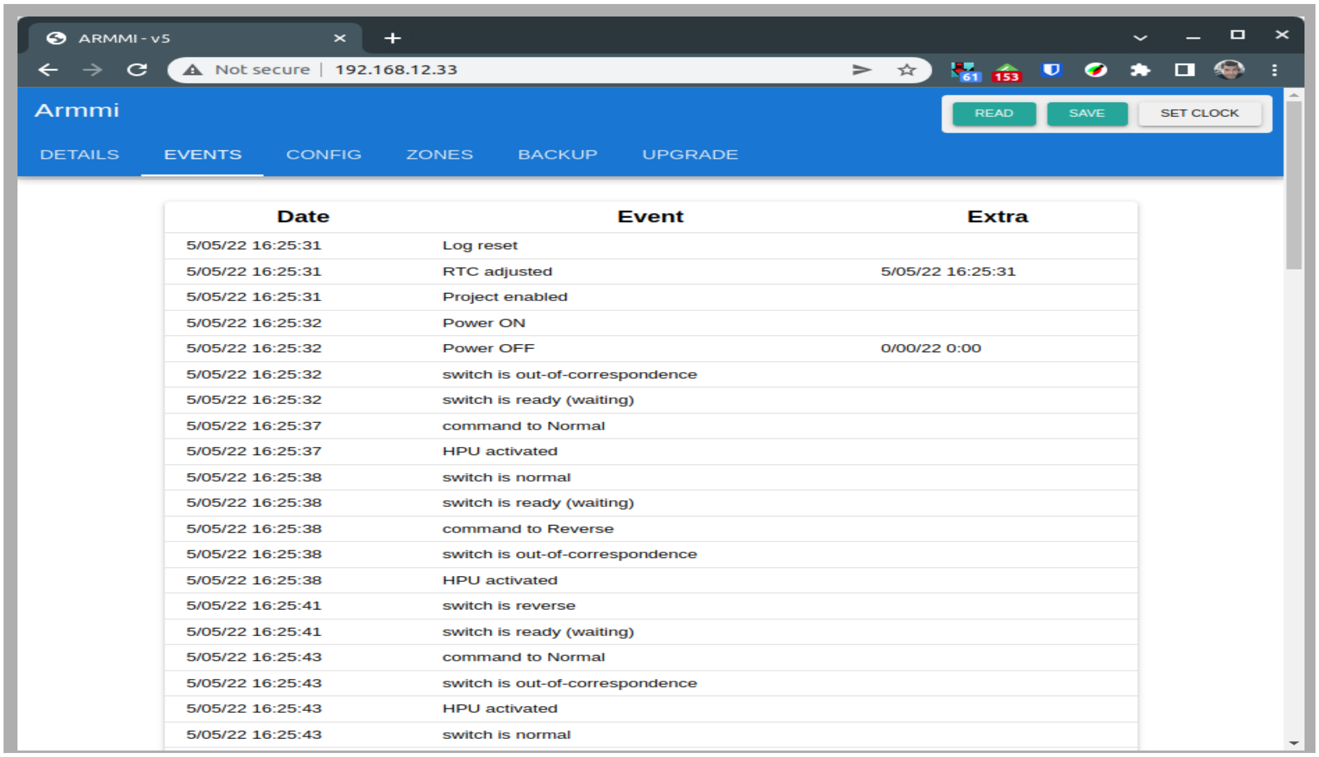
Model	3
Serial Number	1133
Internal clock	2022-05-09 16:39:01
SD Card	Present
Firmware version	V22/125
Firmware Revision	R139
Interface date	Thu, 5 May 2022 12:08:52 GMT
Project description	Single switch _UP_ with dtmf and modbus
Project version	P6
Project date	2022-05-05
Project reference	UP-SW

8.1.1 Events

The *Events* tab will allow to read current unit's events, present them in a table and export them in a CSV format that can be opened by Excel.

It is also possible to adjust ARMMI's clock to match clock of the laptop were the web interface is being used (Usually this is not necessary since this clock syncing is d1 automatically in all upgrade operations.).

Events are mostly defined by installed project. Only a few events like, power on and off, upgrades, clock syncing are managed by ARMMI itself and are always available.



Date	Event	Extra
5/05/22 16:25:31	Log reset	
5/05/22 16:25:31	RTC adjusted	5/05/22 16:25:31
5/05/22 16:25:31	Project enabled	
5/05/22 16:25:32	Power ON	
5/05/22 16:25:32	Power OFF	0/00/22 0:00
5/05/22 16:25:32	switch is out-of-correspondence	
5/05/22 16:25:32	switch is ready (waiting)	
5/05/22 16:25:37	command to Normal	
5/05/22 16:25:37	HPU activated	
5/05/22 16:25:38	switch is normal	
5/05/22 16:25:38	switch is ready (waiting)	
5/05/22 16:25:38	command to Reverse	
5/05/22 16:25:38	switch is out-of-correspondence	
5/05/22 16:25:38	HPU activated	
5/05/22 16:25:41	switch is reverse	
5/05/22 16:25:41	switch is ready (waiting)	
5/05/22 16:25:43	command to Normal	
5/05/22 16:25:43	switch is out-of-correspondence	
5/05/22 16:25:43	HPU activated	
5/05/22 16:25:43	switch is normal	

8.1.2 Configuration

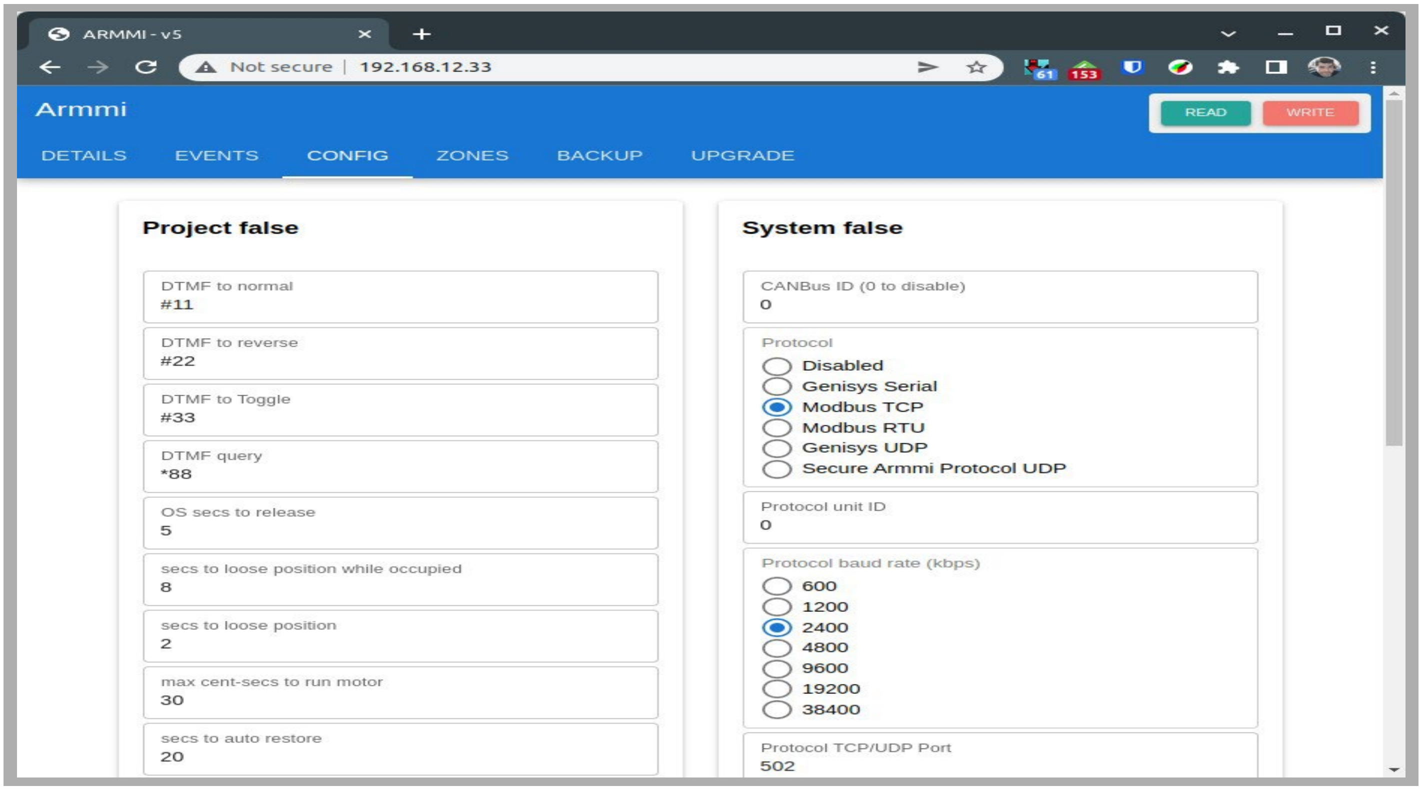
The *Config* tab will read all current configuration and allows adjustment, as necessary.

There are 2 columns of configuration parameters, the left 1 is completely defined by the currently installed project (check Project's datasheet to know their function).

Here you will find DTMF codes, DTMF voice messages and any other configuration adjustment that sets how to Projects will operate.

The right column will present system configurations. Those are defined by current firmware version. Current values will be preserved in between upgrades as long the new project being installed is using the same firmware.

System configuration will allow the user to set ARMMI's communication parameters like IP, TCP/UDP ports, units IDs and port speeds and active protocol.



Project false

DTMF to normal
#11

DTMF to reverse
#22

DTMF to Toggle
#33

DTMF query
*88

OS secs to release
5

secs to loose position while occupied
8

secs to loose position
2

max cent-secs to run motor
30

secs to auto restore
20

System false

CANBus ID (0 to disable)
0

Protocol
○ Disabled
○ Genisys Serial
● Modbus TCP
○ Modbus RTU
○ Genisys UDP
○ Secure Armmi Protocol UDP

Protocol unit ID
0

Protocol baud rate (kbps)
○ 600
○ 1200
● 2400
○ 4800
○ 9600
○ 19200
○ 38400

Protocol TCP/UDP Port
502

8.1.3 Update

Tab *Upgrade* allow to install a new firmware and/or application to this unit.

Applications are created with a custom language named *armmill* (ARMMI's Logic Language), and once compiled they are distributed in a file with extension *armmiproject.* Those files are named with a standard format that will inform project name, description, version and current firmware installed with it. Example:

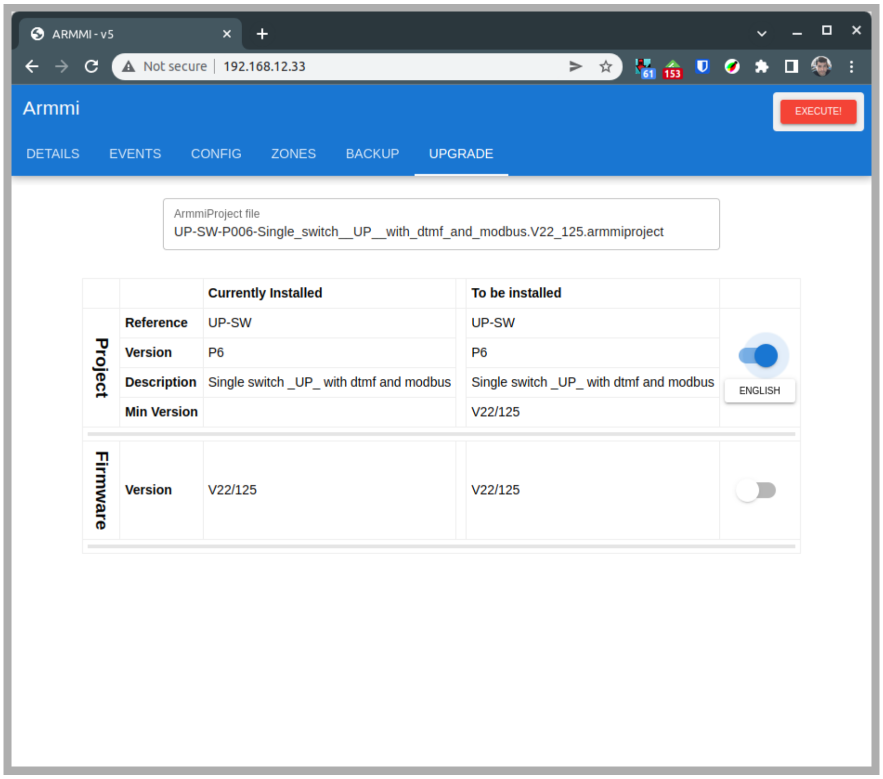
STD-R-SW-P008-Remote_switch_with_aimcan. V22_125.armmiproject

Application reference **STD-R-SW**, version **P008**, description: **Remote_switch_with_aimcan** and firmware **V22_125** is packed inside this same file

Once the *armmiproject* file is loaded at this tab the web interface will show data from what is currently installed in connected unit and data from loaded file.

Applications may have a minimum firmware version requirement. When this happens, and currently installed firmware does not satisfy that requirement, the new firmware will be installed by default. It is up to the customer to upgrade firmware or not (by selecting the switch to the right).

When writing an application, developer can optionally create texts in multiple languages to be able to use same application in different countries. If the loaded application has multiple languages, a selection box will allow the intended language to be selected before upgrade execution.



ArmmiProject file
UP-SW-P006-Single_switch__UP__with_dtmf_and_modbus.V22_125.armmiproject

		Currently Installed	To be installed	
Project	Reference	UP-SW	UP-SW	● ENGLISH
	Version	P6	P6	
	Description	Single switch _UP_ with dtmf and modbus	Single switch _UP_ with dtmf and modbus	
	Min Version		V22/125	
Firmware	Version	V22/125	V22/125	○