

|  |  |
| --- | --- |
| **Document Type** | General Documentation |
| **Project Codename:** | AMBER-PI-PLUS |
| **Version** | 0.9.0 |
| **Author** | Carlo Maria Curinga |
| **State (Draft/Proposed/Approved)** | Proposed |

1. Intro

The scope of this project is to develop a production-ready device that could demonstrate the full potential of resin.io in both the fast-prototyping and light-Industrial field.

1. Background

In resin.io we were able to gather a very wide and valuable feedback from professionals. This assets is critical for the project, since puts resin.io in the position of knowing exactly the market demand.

An interesting finding is that the resin.io platform is ahead of the hardware capabilities: the current hardware offering is holding users from being successful. Main issues are:

* On-board flash absent or too volatile (corruption, etc)
* Bad software support from OEMs (kernel, device drivers)
* Uncertain guaranteed lifecycle and availability
* The need to add complexity to a device in order to satisfy requirements (HATs/Shields “frankenstein” setups)
* The end result of the hardware setup is often off the cost target

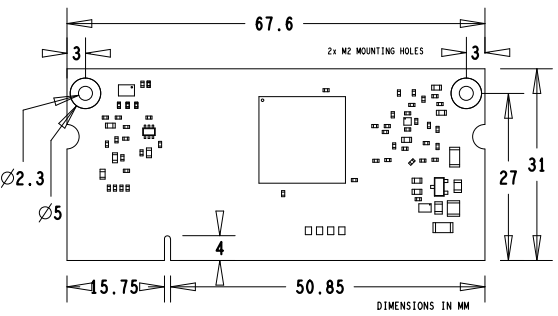
1. Target

Project Amber aims to develop a device offering organized in 3 tiers:

|  |  |  |
| --- | --- | --- |
| **Tier** | **Target Price** | **Description** |
| Starter | 49$ | A simple, reliable platform that covers the basic needs from the maker community in MVP/Fast prototyping phases. |
| **Plus** | **99$** | **A solid and reliable platform with several features embedded (sensors, I/O, co-processor, etc).**  **This tier should be the way to go for prosumers, advanced makers and startups/companies that don’t require aggressive industrial specifications/certifications.. This tier is expected to be the best-seller among the 3** |
| Pro | 199 - 299$ | The Plus with Industry standard additions like:   * RS232 * RS485 * CAN bus * Embedded 3G/4G modem * Industrial I/O connectors * Alu case |

1. The Core

The chosen SoM (System on Module) is the [Raspberry Pi Compute Module 3 Lite (CM3L)](https://www.raspberrypi.org/products/compute-module-3-lite/).



1. The co-processor

In order to allow real-time operations, Analog and high-voltage handling and aggressive power-management, the Amber project will include a co-processor on (at least) the Plus and Pro tiers. The co-processor will be hooked to the Core via i2c.

The following device was chosen for this scope based on price and requirements matching:

* **Samsung** [**Artik 020**](https://www.artik.io/modules/artik-020/)

1. The Carrier: Plus

|  |  |
| --- | --- |
| Availability | At least 5 years since launch date |
| Size | Ideally within 80mm x 60mm |
| Temperature Range | -25 to +80 degrees Celsius |
| Power | * From ~11V to 24V * 1-2 cells LiPo Battery connector with embedded buck regulator, ups and I2C interface (level, status, events) [ suggestion: inspire from the [BeagleBone Blue](https://github.com/jadonk/beaglebone-blue)} |
| RTC | Yes, via co-processor, condenser-based (no coin-cell battery, best suited for industrial certifications and easy shipment) |
| Radio | Cyw4343w (same chip as rpi3) WiFi+BT combo with embedded antennae and uFL connectors for external antennae. |
| Ethernet | 1 x RJ45 Ethernet port |
| Storage | * 8GB emmc hooked to the CM3L SD/eMMC interface pins |
| I/O (CM3) | 40-pin standard [HAT](https://github.com/raspberrypi/hats) connector |
| I/O (Artik020) | I2C, UART, SPI, Analog, protected Digital. Pins exposed with the same type of headers (male pass/through) used for the CM3L, grouped by logic (Power, UARTs, I2Cs, and so on) |
| Sensors | * Humidity + Temperature (via Artik020) * Accelerometer + Gyroscope (via Artik020) |
| User feedback | 1 x RGB LEDs (via Artik020) |
| USB | 2 x Female type-A 2.0 |
| Video output | * Micro HDMI * DSI DI5 (rpi display connector) |
| Video input | 1 x CSI camera connectors (rpi camera connector) |
| Expansion | Embedded mPCIE controller |
| Debug | Serialport exposed via HAT headers |
| Boot | micro-USB slave port with switch (same as CM3 evaluation board) |

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### Expected Use Cases (general)

* Device acts as a ROV / Drone logic unit, controlling sensors, motors (usually 6V+, brushed and brushless vie ESC ) powered by 2-cells LiPo batteries
  + Co-processor takes care of basic functioning like stabilization, flight/flow controller and motors/sensors abstracted interface for the core (CM3L)
  + Core takes care of high-level edge-computing tasks like data processing, machine vision, etc
* Device acts as a stand-alone data hub device, powered via solar panels with a 1-cell LiPo battery backup
  + SCENARIO1: Co-processor is always on, gathers data from sensors, every X hours boots the Core, passes data to be sent to the cloud and shuts it down again with an ACK-based protocol
  + SCENARIO2: Core gathers data from an hybrid set of sensors (some exposed via Co-processor, some connected to it directly) every X hours, sends data to the cloud and asks the Co-processor to shut itself down scheduling a boot in the next X hours when done
* Device acts as a Digital Signage gateway, reproducing web/multimedia interactive content on a screen, while managing sensors and actuators like proximity, cameras, LED strips, etc.
  + Co-processor handles analog and heavy interfaces like plain LED strips (12V) so user doesn’t need to add logic converters or additional shields to the board
  + Core is always or often connected to the user’s backend, syncing content and analytics, while exposing an HMI solution. It asks the Co-processor to shut itself down (and ideally the display via HDMI-CEC or RELAY) during the closing hours of the venue and boot itself up again at opening.
* Device acts as main/controlling unit for sophisticated machines like 3d-printers, laser-cutters, harvesters, CNCs, looms, etc
  + Co-processor acts as the mechanical controller - all the business critical logic is handled by itself (ie in the 3dprinter case, it controls the stepper motors an all the parts) - the machine still makes sense and works (but not in a smart way) if you reduce the setup to the co-processor only.
  + Core acts as an abstractor/extender for the Co-processor by adding functionality on top of it like video feed, remote management, diagnostic, etc ( in the 3dprinter case, the Core adds Wireless control, remote files upload, print video feed and queue management)

### Ecosystem Use Cases (amber+resin added value)

...