

---

# Telemetry Documentation

*Release 2.0.0*

**Remi Beges**

December 14, 2016



<b>1</b>	<b>Indices and tables</b>	<b>3</b>
----------	---------------------------	----------



The `Telemetry` project is a collection of tools for *remote data-visualization and communication* with embedded devices.

At its core, *Telemetry* is a communication protocol.

Data is exchanged on named channels, called *topics* (ex : *foo*, *bar* and *qux* on the figure above).

Sending data is called *publishing*.

```
Telemetry TM;
int32_t i = 123;

TM.pub_i32("foo", i);
```

Receiving data is called *subscribing*. A variable attached to a topic will be updated each time new data is received under this topic.

```
Telemetry TM;
float throttle;

TM.attach_f32_to("throttle", &throttle);

for(;;) {
    TM.update();
}
```

Extra tools are build around this protocol to provide a fun and easy way to communicate with the device from a computer in real-time.

The Telemetry project is constituted of different libraries at the moment:

- **Protocol implementations**

- `Telemetry`: Portable C/C++ implementation of the protocol. Runs on embedded platforms (Arduino, Mbed, etc.).
- `Pytelemetry`: Python implementation of the protocol. Runs on Windows, Mac OS, Linux

- **Desktop Tools**

- `Pytelemetrycli`: Smart command line interface. Open **live plots** on received data from a single command,  
reconfigure device in real-time, logging, network analysis, etc. Runs on Windows, Mac OS and Linux

Contents:



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`