

Relay Race Implementation Overview

Definition

A RelayRace is a Race “made of” two or more Internet-connected physical OpenLEDRace Devices. Cars starts in the first OLR, made ‘n’ loops, disappear from the first OLR and appear back in the next OLR.

You can get an idea of what a Relay Race looks like in the video below:

http://www.openrelayrace.org/storage/app/media/RelayRace_1.mp4

Notes about the video:

1. You see two Racetracks (Open LED Race Device + LED Strip).

There are **two** Network Clients (computers) managing **one OLR Device each**.

They exchange MQTT messages via an "external" MQTT broker (*i.e. the Broker is **not** running on a server in the local network*).

Messages from one OLR to the other travel through the Internet to reach the broker and come back.

This means:

1. *The ‘delay’ you see between the moment when a car ‘leave’ a circuit and ‘enter’ the other would be more or less the same if one of the circuit is in a different place (city, state, etc). The delay in the video is around 385 mSec.*
2. *You can use a **local MQTT broker** and play Relay Races within ‘Teams’ of users in the same phisical space. In this case you don’t need an Internet Connection and the delay is around 75 mSec (tested using a local Mosquitto Broker).*

You can see a video of a Relay Race with 3 racetrack and a Local Mosquitto Broker following the link below:

http://www.openrelayrace.org/storage/app/media/3Tracks_noCountdown.mp4

2. There are no ‘Players’.

The code on the OLR Boards is running in “Simulator mode” (*special mode useful to develop/debug the Network Client*).

The software is the same one running on the Arduino when actual players are using the device. The only difference in ‘Simulator mode’ is the software ‘click’ the Player button by itself (*while in “Normal mode” this is done reading the state of the I/O pin associated with the Player*)

Implementation

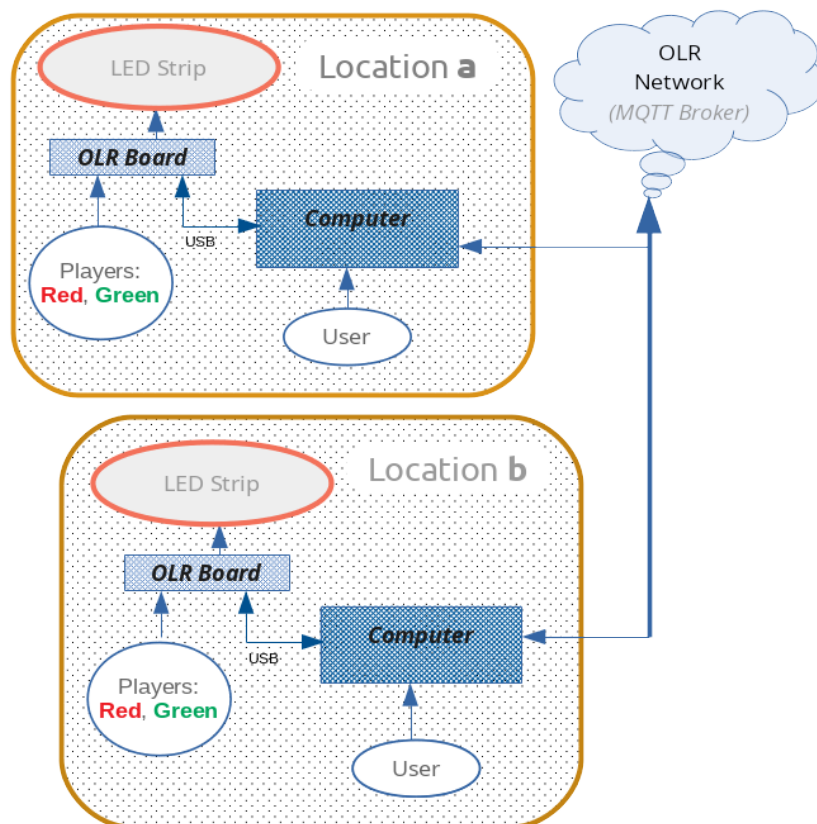
The system is implemented with a **Computer** and an **Arduino** on each participating node.

Computer: Race Configuration, Network

- **User Interface:** *Participants create and join the race (and define race settings)*
- **Network:** Take care of the communications with remotes OLRs, both for tasks running on the Computer itself and for the Arduino.

Arduino: Players Input, Car Simulation, Led Strip

- Implements an “extended version” of an Open LED Race where everything works as usual, but the cars ‘leave’ and ‘get back’ to the Racetrack after ‘n’ Laps. The software manage this exchanging messages with Computer (Network)



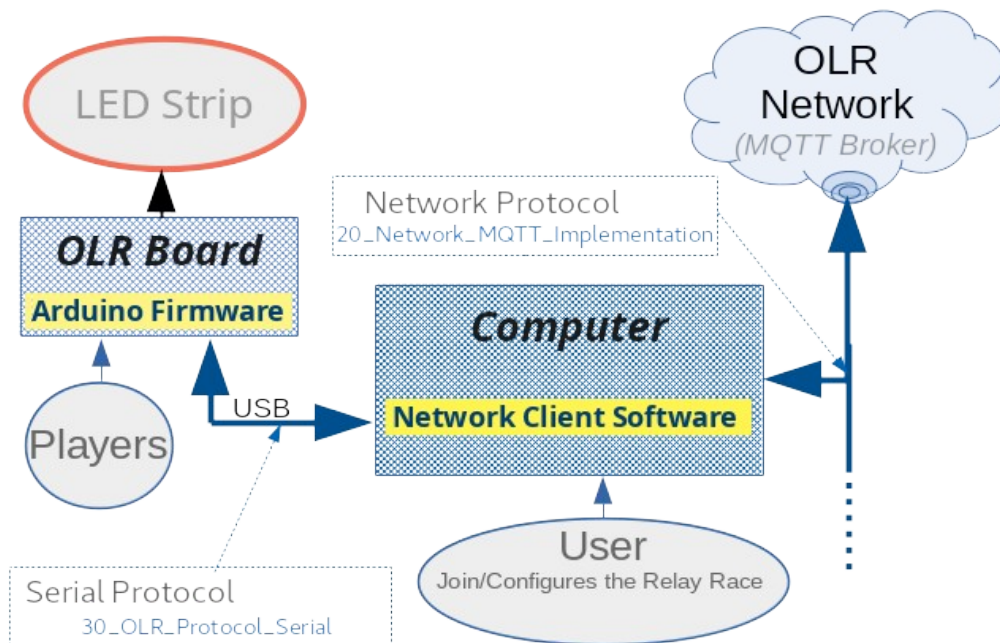
Node

Computer: Network Client Software

- **User Interface:** User can 'see' others OLR currently online and 'Setup' relay races with them.
- **Implements OLR Network:**
 - Connect to an MQTT Broker and PUB/SUB to a set of predefined Channels to get the list of OLR currently online and the list of Available Races.
Implements **OLRNetwork Protocol** (*defined in **this** document*).
- **Local Device communication:**
 - Exchanges messages (via USB/Serial) with Arduino firmware.
Uses a **Local Message Protocol** defined in this document.

Arduino : Arduino Firmware

- Manage Physical Input device used to play (push-buttons, etc)
 - Speed/Position calculation
 - Drive Led Strip
- Exchanges messages (via USB/Serial) with Arduino firmware.
Uses a **Local Message Protocol** defined in this document.



Computer ↔ Arduino interaction

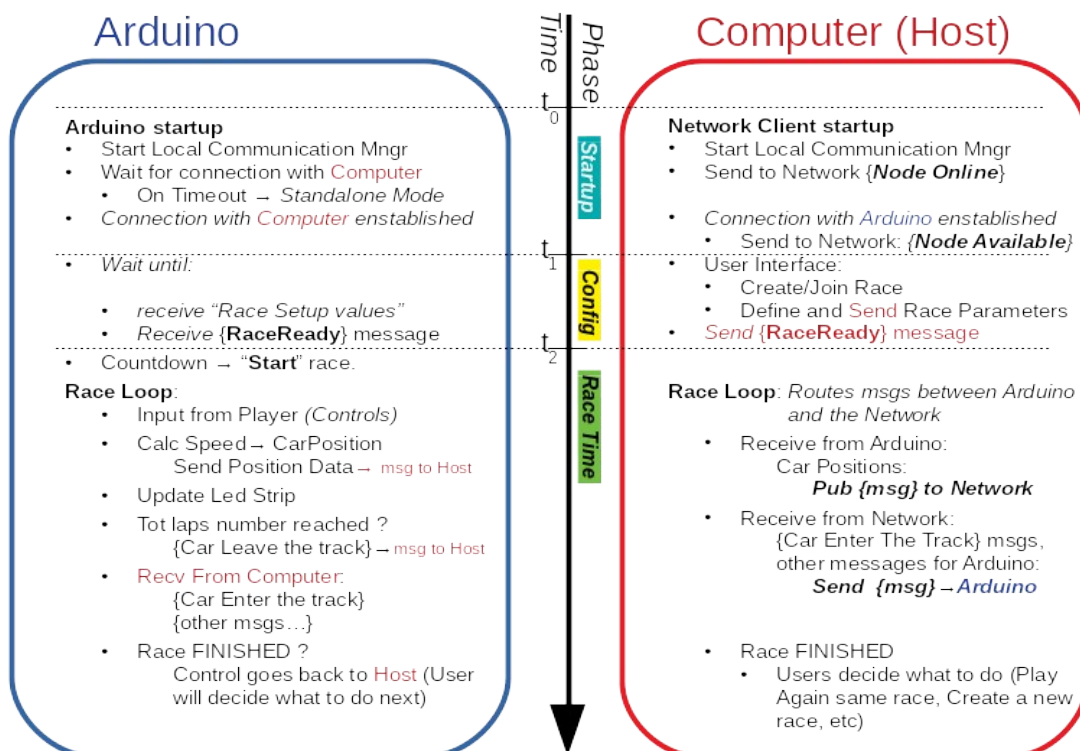
The image below describes the Computer-Arduino interaction during a Relay-Race for one of the participants (for example “Location a” described in a previous page)

The interaction between the software running on the Computer and Arduino may be divided in three phases:

- **Startup**
 - User turns on Arduino and launch the Network Client on the computer.
 - The two software execute their initializations and perform the handshake
 - The Network Client updates Node’s status on the Network (MQTT Broker)
 - Take a look at this [simple visualization of the Startup Phase.](#)
- **Config**
 - Computer: The User create/join/configure a relay race with the Network Client software (*Computer*)
 - Arduino: wait until the user ‘confirms’ the configuration and the Network Client sends it to the Arduino
- **Race time**
 - Arduino: manage the race (read game controllers, update cars position in the LED strip, etc) and exchange commands with the Network Client (car A leave the Racetrack, etc)
 - Computer: The Network Client ‘routes’ commands between Arduino and the Network

Please Note:

In the description below, the Arduino manage the Start and the Finish line of the Race. The “remote” OLR (Remote Arduino) will receive the cars after ‘n’ laps and send them back after ‘m’ laps.



Software Modules

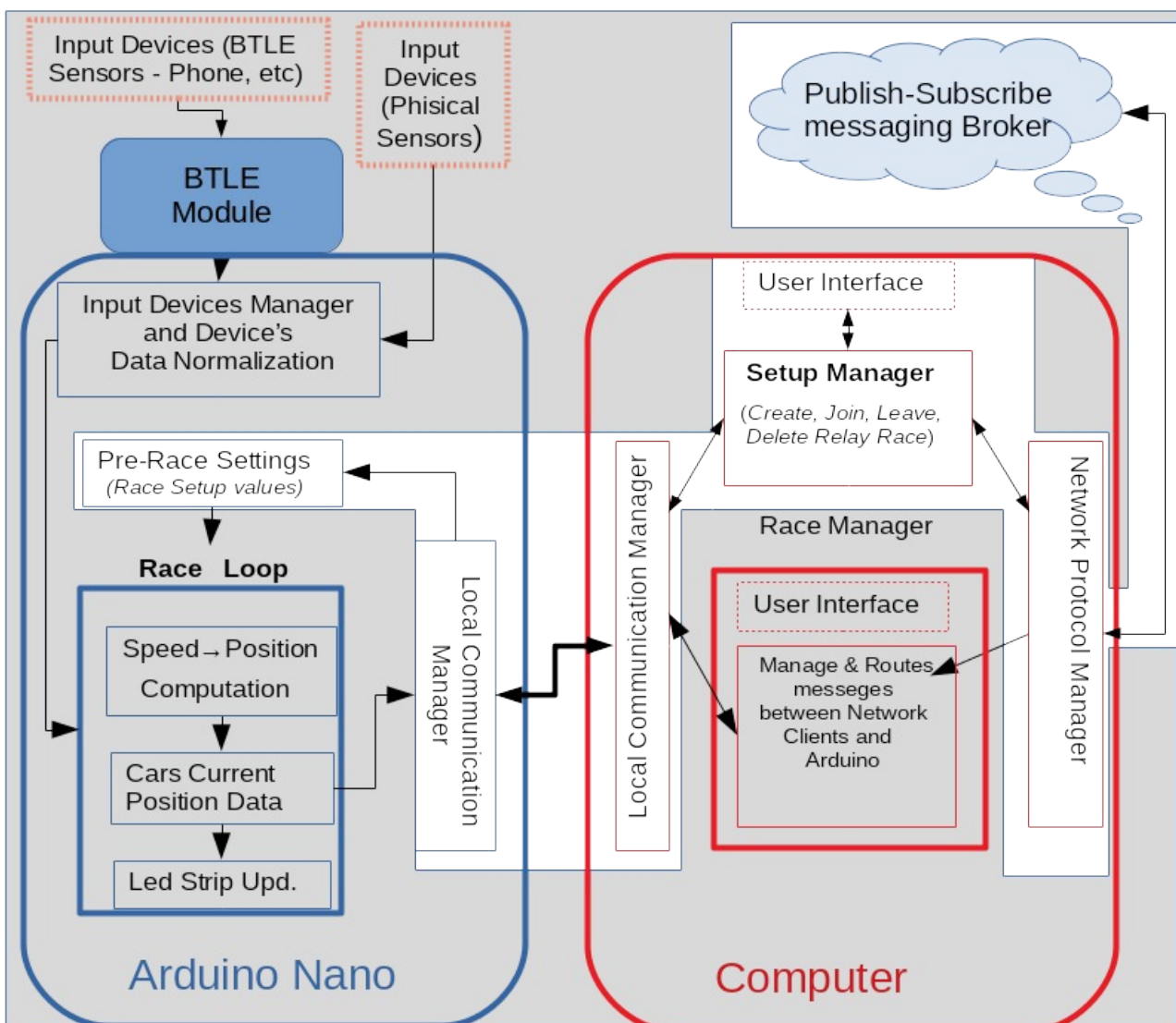
The image below is a visual representation of the software modules for the whole system. The parts not greyed-out are the ones involved in the process of "Creation" of a Relay Race (first thing the user does when he/she wants to play a Relay Race with one or more OLR currently online)

This task is implemented in the [Setup Manager] program, running on the Computer

Typical usage

1. User at Location (a) creates a New Race.
2. Another user, at location (b), Join the new race.
3. User at Location (a) [Configures] the race (set race configuration parameters):
 1. Order (1st Track - where race starts, Second track,...last one where race ends).
 2. Laps in each device
 3. ...etc
4. When done, the Config Parameters are sent to every single Client and Race go to state [Ready]

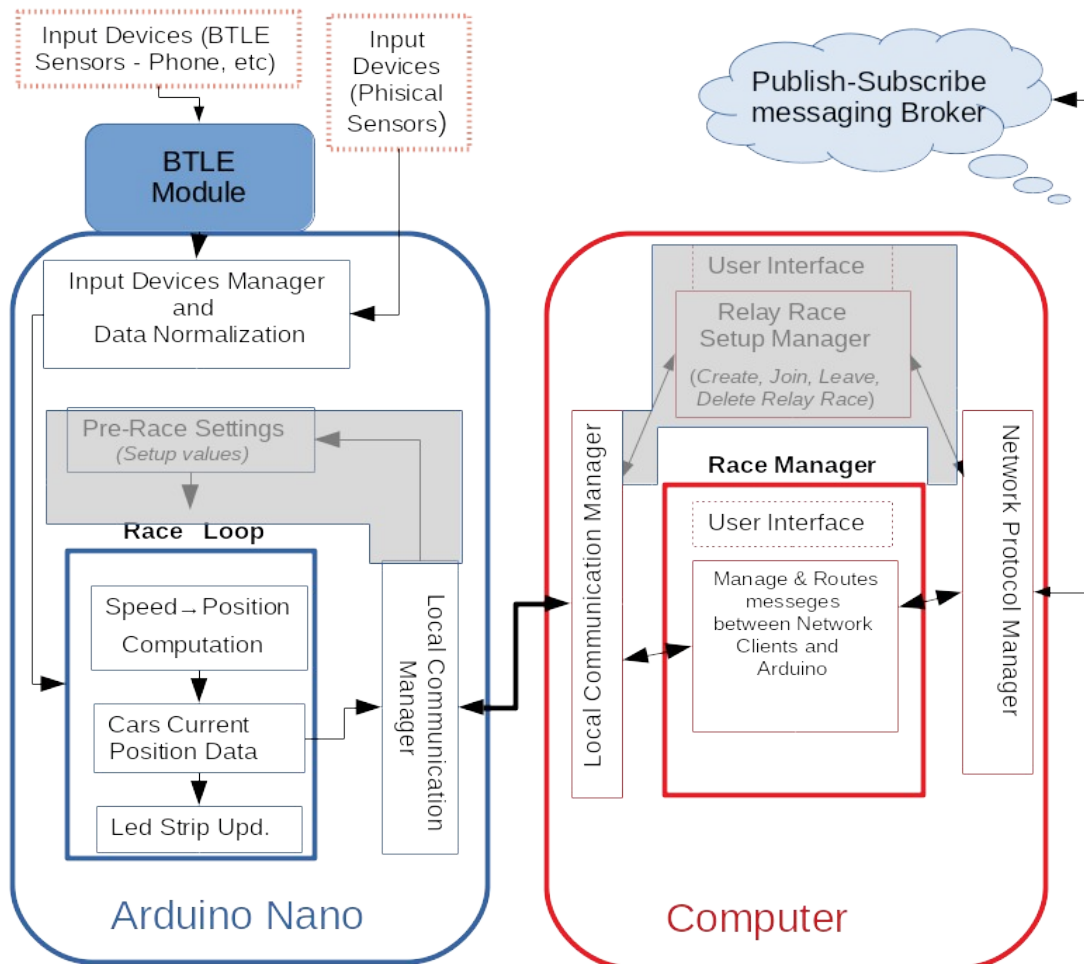
Each Network Client, on receiving the Setup Values, send them to the Arduino (Race Setup Values) and the [Setup Manager] module leave the control to the [Race Manager]



Race Time Phase

The image below shows the modules managing the “Race Time” phase.

If a car is in the Local Racetrack, the player “*drives*” the car as fast as he/she can. If the car is currently in another racetrack, the player is looking at the screen, waiting for the car to come back here (and Arduino is waiting for the “Car Is Here” message from the Network Client)



Notes

Arduino

- **Setup Values**
 - Local hardware setup
 - LED Number in the Strip
 - Gravity, Slope, etc
 - Relay Race Config (Network)
 - RacetrackPosInRace (1st, 2nd,..last)
 - LoopsN, etc
- **Input Devices Mngr and Data Normalization**
 - Input devices may be very different (Buttons, Force Sensors, Bicycles...).
 - This module manage data input from different sensor and **normalize** the output for all of them in a fixed range (ex: 0-1000)
- **BTLE Module**
 - Interface to connect BlueTooth sensors.
 - It enables, for example, the usage of a mobile phone as a Input Sensor (Ex: shake, flip, push a button,...)

Computer

- **User Interfaces**
 - The UI for [Setup Manager] and [Race Manager] may use a display directly connected to the Computer or implemented to be used remotely (ex: via Browser)
- **Setup Manager**
 - Let the user create or join a relay race, and define the Race's attributes (where race starts, where it ends, how many laps, etc)
- **Race Manager**
 - During the “Race Time” phase, the software ‘routes’ messages between the local Arduino and the Network .