Counting Passes in a Corridor Project

Report

Martin Roznovjak, Roni Selin and Niruj Khadka

Group 1 and 11, IoT Project TX00CP85-3003

Academic year 2017/2018, period II

Idea of the project

We wanted to combine several technologies we came across during the semester, get deeper insights in working with them and doing it in a relatively short time frame without it being overly simplistic. We had an idea that it might be interesting to see a histogram of people passing through a corridor in each direction.

Plan

We developed a plan where an Arduino Uno microcontroller would be powering and reading two proximity sensors near each other along the corridor (so that we could tell the direction based on which one was first triggered, but close enough so that it would work properly when more people are passing simultaneously), the alternatives were a capacitive sensor (which turned out to be very noisy) and a pressure sensor on the floor (which we did not make nor got access to). The Arduino Uno would take care of the sensing logic and would communicate it serially over to a Raspberry Pi which would be running a database keeping records of the events received from the Arduino Uno and for real-time publishing of the events it would be connected to the Internet, either also running an http server hosting a publishing website or exposing its database to this website over an SSH tunnel for security.

Implementation

During the implementation we faced problems mainly with our IR proximity sensors - we figured out that using a reflexive tag opposite of a sensor would prolong its range sufficiently, however, one of the sensors stopped working very soon, we tried building a new sensor using a pair of an IR transmitter and a receiver, but as they were lacking documentation we did not manage to get them working. We noticed that a fire sensor was designed to ‘see’ a fire using an IR receiver, which turned out to be sufficient for our purposes. We turned the proximity and the fire sensor into IR receivers and were using an IR LED to form a beam between them (as of only having one such LED we had to make the arrangement triangular instead of rectangular - one source, two receivers). However, we were still lacking the documentation for the IR LED and the range for the sensors was only up to about 70 cm, while trying to increase the range, the IR LED burned out. We could not get a new IR LED soon enough, so we used ordinary candles as a source of IR light together with an aluminum foil as a reflector to direct the radiation. Surprisingly, the set-up was working reliably and we wrote the sensor-logic code accordingly, however what we forgot to realize is that the source of the IR beam was much broader than when using the IR LED, that has a consequence of using different timing constants in each implementation to filter false positives and check for errors; on the day of our presentation we had a new IR LED to use and we had not realized this issue, thus the sensor was not working flawlessly during the demonstration.

The communication between Arduino Uno and Raspberry Pi was done over USB and using open source serial libraries, there were no struggles. We deployed Raspbian Lite operating system on our Raspberry Pi, enabled SSH and were further configuring it over SSH connections. We installed and configured a MariaDB (a MySQL fork) server together with a database for keeping the records (we record four types of events together with the time when they occured: forward and backward direction of passing, error condition of the sensor and a reset event - when it started sensing after bootup or an error state) and user accounts for reading (website) and writing (serial communication daemon) from/to the database.

Due to the problems with the sensors, we lost a lot of time and did not have enough time to create a website with statistics and good user experience to publish our data, we only sketched a webpage which connects to the database and shows latest entries in the database using PHP. We also decided to have the HTTP server directly on the Raspberry Pi utilizing Apache2.

The sources codes can be found at https://github.com/Avatust/uno-raspberry-dollop as most of the configuration was done 'on-the-fly' the configuration files are not included in the repository, the database and its accounts were created manually using MariaDB's interactive prompt. Some photos are included at the end of this file.

Conclusion

There is plenty of room for improvements in this project - whether it is the presentation of the data or its functionality and versatility, or overcoming limitations (this setup will take two people walking side-by-side as a single person only).

We learned that putting different technologies together does not have to be a big struggle, we especially learned that we should reserve much more time for unanticipated problems. In future projects, we would like to improve in our team-working - to specify our roles better and come quicker to conclusions and understanding each other and decrease our overhead in getting started. Overall, we enjoyed working on this project very much.