Information system for time measurement

Andrej Kováč

Abstract:

In this article a combined system for measuring times on different motorsport events is presented. This system aggregates two main information technology groups – hardware and software solutions. From hardware – it is Raspberry Pi, Outdoor display (output), Optical photosensors (input). From software – Application stage (Java), Database (MySQL), Webpages (PHP + HTML + JavaScript + CSS). All named parts are connected to one complex system for measuring, saving, processing and displaying data.

Keywords:

Time measuring, car races, hardware, software, display, sensors.

ACM Computing Classification System:

Sensor applications and deployments, information systems application.

Introduction

Time measuring or time keeping is a regular part of our lives. Everyone have some clock, watch, etc., because we need to manage many things, and without any time orientation it is often impossible.

From history, the sexadecimal (60) system of time measurement was founded around 2000 years BC. Then through the years there were many kinds of different time-measuring devices. For example candle clocks, sundials, hourglasses or astronomical clocks. Time measuring is closely connected to many sport events, which must be measured for best results [3].

This project is linked to the sport sphere too. Most of motorsport events have to be measured – rally, circuit racing like Formula 1, amateur time-attack racing, car slalom, autocross or hill climb racing.

Measuring time in my project contains car slalom, amateur time-attack or hill climb racing, where cars start and finish one by one and the number of cars on track is less than 10. But there is no problem to increase this or other numbers in this project if necessary. Our local system for time measuring has just few functions: max. 2 cars on track, connection with optical gates and external display, showing measured times to steward. The main aim of my project is to introduce a complex system which collects data from sensors and store it in the database, or show it as live data for stewards or race spectators. For better orientation, there are not only numbers, times and race results in the database, but names, cars, and also other driver information. For this purpose, there is a registration webpage, which can be completed by a steward or a driver prior to the race. Very important is the overall resulting, which usually varies in different races. For example - on time-attack race are 3 measured times, but only 2 best are sum-up to the result. In hill climb racing are 2 practice times summed up and separately 2 “sharp” times sum up together. The final part of the system is showing race results and live data to visitors through the webpage or external display which allows them to be part of the drama if there are some fights between the drivers, and the visitors have immediate information about times, when the cars cross the finish line.

1 Hardware components

The information system contains as well as software, hardware parts. All their technical features and functions are described below.

1.1 Raspberry Pi 3 B+

The basics of information system is a small single-board computer [1] from Raspberry Pi Foundation (Figure.1). It is the third generation of Raspberry models a with mark B+.

Figure 1. Raspberry Pi 3 B+ [2].

Main benefits are its size, power and reliability. Raspberry offers few operating systems like Raspbian (Linux), Ubuntu mate, Windows 10 IoT core, Pinet or Risc OS. The best one for this project was Raspbian, which offers GUI and many other features:

- SoC (System on a chip): Broadcom BCM2837

- CPU: 4x ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: micro SD
- GPIO: 40 – pin header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

1.2 Optical gates

The source of input information about a car if the start or the finish line were crossed, are optical sensors. One optical gate contains a transmitter and a receiver (Figure.2).

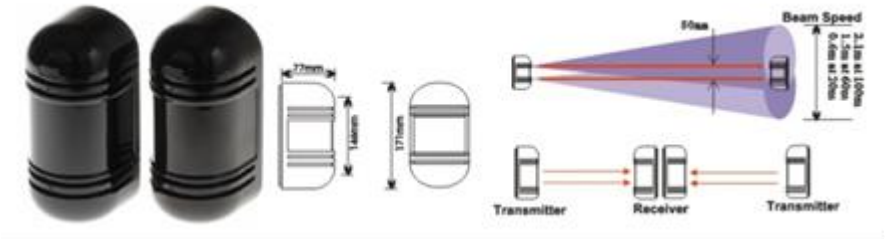

Figure 2. Optical gate – transmitter + receiver [4].

One gate is situated on the start line, the second one on the finish line, where both are placed on each side of the line/road. The transmitter transmits a beam, which is received by the receiver on the other side of the road. When the beam is crossed, the relay in the receiver connect two pins together – +5V and input pin of Raspberry Pi. For transferring data from gates to Raspberry Pi can be used cable or connection through the air. We use communication set (transmitter and receiver) from ERC SportDisplay company for wireless connection. This set is based on AUREL TX-SAW 433,92 MHz transmitter module [8]. With two direct antennas on transmitter and receiver we can set wireless connection on 2 km distance. Optical gate and transmitter for sending data can be placed in this distance with powering from 12V battery (for example car battery).

1.3 Outdoor display 7 segment

All the race visitors want to see the driver's final time immediately after he crosses the finish line. Good placed outdoor display is the easiest way to show this information. The size of this display is 185x70 cm and there are two rows for two measured times of drivers (Figure.3).

Figure 3. Design of 7 segment outdoor display, from left to right: rank, starting number, category, time, round, penalisation.

The display is made by ERC SportDisplay company and one row contains this information (from left to right): actual driver place from results, starting number (id) of the driver, category in which driver belongs, measured time, penalisation and at the top is the current lap. These 7 segments are controlled by shift registers which were a part of the display [5]. Input voltage is only 12V what is a big advantage, because if there is no 220V electrical system, the display will run only from a car battery for many hours.

1.4 Other hardware

For a control application of the information system which runs on Raspberry Pi, we need a PC mouse, PC keyboard and display. It is a default control, but raspberry supports VNC (remote control) so there is no problem to control everything from a laptop, tablet or a smartphone. We can also share the results with the visitors through the wi-fi network which Raspberry can setup. When a visitor wants to see the race results, he can connect to the wi-fi network and he is automatically redirected to the result website on Apache HTTP server. Internal wi-fi module of Raspberry Pi can be enough for testing, but for more power, and maybe more than 100 visitors the external router is needed.

2 Software components

All software components are described below. LAMP software was installed on Raspberri Pi which includes the Apache HTTP Server, the MySQL relational database management system, and the PHP programming language. The main application was written in Java programming language.

2.1 Java application

Application is intended only for the steward which controls the start of the cars which time is being measured. This steward must have an eye contact or a contact with stewards on the track if some car didn't crash. All logic for measuring times, is programmed in this Java application and it is very simple [6].

When a car crosses the start line, the current time is stored in database. The same happens when a car crosses the finish line. The final time of this car is a difference between finishing time and the starting time (result = finish time – start time). The main functions of the application can be divided into two main groups: background and foreground.

On background, pin control, through the pi4j library, is very important. Two pins are inputs for optical gates and 5 pins are outputs for sending data to the external display. Data from inputs are taken to listeners, which start or stop different actions. Data to be shown are sent to the display in a separate thread, by shift-out function, which contains bit operations.

On foreground runs Java JFrame application for showing all important data to steward as live-measured times of the drivers on the track, their names, car, start number, actual lap. Steward can setup a start number of the driver who is ready to start, actual lap, change position of drivers (if one car overtakes another car), or remove some driver (if one of the cars on track spoils). Measured times are sent to database, and then shown through the webpage which allows the steward to correct times if a car has some penalisation, or another problem.

2.2 MySQL database

All the results, times and other data about drivers are stored in MySQL database. Before race, the steward fills in the registration form with a category, start number, name and a car of every driver. Then the race and the time measuring can begin. When the driver comes to the start, the steward checks his start number and marks it in the application as "Car on start". When the car crosses the start line, the current time is uploaded to the database. The same happens when the car crosses the finish line. These results are shown to the race visitors on competition through the webpage, where tables are generated, including: driver's name, car, category and result time from chosen lap.

2.3 PHP + HTML webpages

Upload data from a form or show data from MySQL database is optional with PHP webpage [7]. Registration page is an easy form page which includes a dropdown menu, text inputs and submit button. After the data was sent to the database, the updated table is shown below the form. Another webpage is only for showing results for the visitors. The third webpage is similar to the previous one for showing results, but it is modified only for the steward to change or update values in the database. For example, if some car has penalisation, the extra time can be added to a special column. This PHP webpages runs on Apache HTTP server installed with LAMP software.

Conclusion

The main aim of this work is to introduce one complex system for measuring times on different motorsport events (Figure 4, Figure 5). Hardware solutions, which are necessary for a connection with the real world, are described here and working with them means to study not only theoretical but practical things too. Connecting all components required knowledge of microelectronics components used in outdoor display or optical gates.

Software solutions is the second part, which gives soul to all hardware components. This project demonstrated a connection between several IT spheres. The main application is written in Java, MySQL database, PHP webpage with HTML, JavaScript and CSS code or scripts.

This system is further developed with testing on race practise in real conditions.

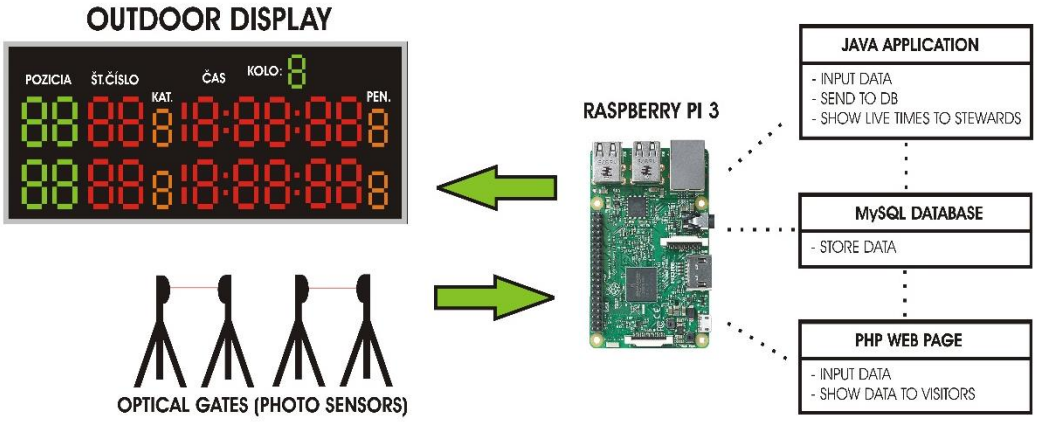

Figure 4: Complex time measuring system

Figure 5: Local car race, current display design

Acknowledgement

This work was supported by my mentor Ing. Juraj Štefanovič, PhD., who helped me develop this system and gave me many consulting hours. Also thanks to Ing. Ján Doboš and Ing. Tomáš Páleník, PhD. for consulting parts of the system, and Paneuropean University for supporting me during this project. I am also thankful to ERC SportDisplay company for support with hardware components – outdoor display and optical gates.

References

[1] Raspberry Pi official site, Available online: https://www.raspberrypi.org/
[2] The Pi4J Project, pin setup for Raspberry Pi 3 B+, Available online: http://pi4j.com/pins/model-3b-rev1.html
[3] Barnett, Jo Ellen, Time's Pendulum: From Sundials to Atomic Clocks, 1998, San Diego, CA: Harcourt Trade Publishers. ISBN 0-15-600649-9. OCLC 40255897
[4] Sunwave, Photo electronic beam sensor documentation, Available online: http://www.securityone.com.sg/uploads/products/product_774270840.pdf
[5] 8-bit parallel-out serial shift register, Texas Instruments, 2014, Available online: http://www.ti.com/lit/ds/symlink/sn74als164a.pdf
[6] Getting Started with Java® SE Embedded on the Raspberry Pi, Oracle, August 2012, Available online: http://www.oracle.com/technetwork/articles/java/raspberrypi-1704896.html
[7] PHP Documentation, Copyright © 2001-2017 The PHP Group, Available online: http://php.net/docs.php
[8] Aurel TX-SAW/433 s-Z Instruction manual, AUR°EL S.p.A., 2001, Modigliana (FC) ITALY Available online: http://ottomat.hu/RFmodulok/650200689_um.pdf

……

Bc. Andrej Kováč
Faculty of Informatics, Paneuropean University in Bratislava
AndrejKovac2828@gmail.com