An Autonomous Bicycle

GUI, Database, Connectivity

Christian Kreipl

September 25, 2017

Contents

No previous work to built upon. Everything has to be done from scratch.

1 Goal

2 Choice of mobile device

At first we are supposed to choose a suitable mobile device to display data, send commands to the bike and display recorded data. To be able to decide on a certain device, we need to specify its requirements. We consider the following points:

- Display size and resolution
- Connection options
- Operating system
- Input type

With respect to:

- Mobility
- Usability
- Effort to maintain and setup
- Price

To rate the options we introduce the rating system shown in table 2.1.

Table 2.1: Ratings and weights

Very positive	Positive	Neutral	Negative	Very negative
+ +	+	0	-	- -
Very important	Important	Neutral	Unimportant	Very unimportant
VI	I	0	U	VU

We rate with and without respect to the price. In the latter case the price is matched to a reasonable device. E.g. if there is a 600€device and another device, that fits our purposes, for 200€, the cheaper device would be chosen. Usually there are cheaper devices available then the minimal price that is stated in the following tables. This devices are usually to weak to be considered as relevant candidates and therefore left out.

Table 2.2: Display size and resolution

	Small ($< 5''$)	Medium ($5'' - 10''$)	Large ($> 10''$)	
Usability	fairly hard to see details	easy to see details	easy to see details	VI
Price	80€+	150€+	200€+	I
Resolution	up to 2160p	720p - 2160p	up to 4K	U
Weight	low	low - medium	medium - high	0
CPU	low - medium	medium	medium - high	I
Rating w/o €	-	0	+	
Rating with €	-	+	0	

The "small" devices are smartphones, the "medium" sized devices are usually tablets or netbooks and the "large" devices are laptops. Desktop PCs are ruled out because of their lack of mobility. The smaller the device is, the higher is the mobility of the user, but also it is problematic to display a lot of data at the same time and allow control. The price scales with the computational power of the device.

Table 2.3: Input devices

	Touch display	Mouse & keyboard	
Mobility	high (smartphone, tablet)	medium - high (laptops)	I
Usability	high	high	VI
Price	0	0	U
Effort to setup and maintain	high if OS $\neq$ Android, otherwise low	low	VI
Rating	+ if OS = Android, otherwise - -	+	

In the considered devices the input type is already built in. Therefore no additional costs are added. This applies to other parts aswell. For instance are almost any mobile devices equipped with WLAN and Bluetooth. If a touch display is chosen, the operating system should be Android (iOS devices are generally to expensive for our purposes).

Table 2.4: Connections

	WLAN	Mobile network	Bluetooth	
Range	85m	global	5-10m	I
Price	0	2x 10€	0	0
Volume	∞	1GB Data / Month	∞	I
Latency	1-80ms	GSM: latency 500 ms+ EDGE: latency 300-400 ms UMTS: latency 200 ms LTE: 35-40 ms (advertised with 10 ms)	40-250 ms (40ms was the lowest latency advertised)	I
Effort to setup and maintain	low	high	high	VI
Rating w/o €	+ +	0	-	
Rating with €	+ +	-	-	

The best option for the connection type is WLAN. It has a sufficient range, is free is fairly easy to use and maintain. It is assumed, that the bike is in an area that is free of disturbing signals. The bike supports both 2.4 Ghz and 5 Ghz. However it is not able to open a 5 Ghz hotspot due to legal restrictions. The WLAN adapter is not region locked and therefore can't open an access point. It can however connect to a network that is spanned by a router or the mobile device. Bluetooth would only be an option if the usecase would be restricted to someone moving in close distance to the bike, e.g. riding on it. The global range of the mobile network comes at the price, that a mobile device must be attached to the bike be able to open the connection. This increases the risk of failures, limits the data that can be sent and is a huge effort to setup. In the current usecase the global range is not even useful, as the bike is not expected to move far from the controller.

Table 2.5: Operating system

	Android	Linux	Windows	
Usability	good support for touch, allows easy access to sensors	possibly bad driver support	usually good driver support	VI
Price	0	0	0-280€	0
Effort to setup and maintain	high	low	low	I
Rating w/o €	0	0	+	
Rating with €	0	0	0	

The operating systems are all fairly equal with their own strengths and weaknesses. None of them has a huge advantage against the others. A touch display probably has the

best support under Android and Windows 8. The additional access to the built in sensors would make the effort to set it up and get an application running valuable. If no special hardware is required, Linux provides an open environment and a lot of freedom. The operating system should therefore be chosen with respect to the underlying hardware.

Table 2.6: Final comparison

	Android	Linux	Windows
Smartphone	-	N/A	-
Tablet	+	+	+
Laptop	0	0	0

3 GUI

4 Database

5 Communication

6 Student project