Bicycle ABS

Embedded Systems

Technical University of Crete
Winter Semester 2017-2018

Team: Kritharakis Emmanuel, Fotakis Tzanis

emails: kritharakismanolis@gmail.com, fotakistzanis@gmail.com

GitHub: https://github.com/TFotakis/Bicycle-ABS

Milestone 3 - Bicycle ABS

Introduction

Purpose of this project is the creation of an embedded Anti-Lock Braking System (ABS) for bicycles using an AVR microcontroller.

In general, ABS, like what its name stands for, is the system that is responsible for avoiding the locking of any wheel while braking. The whole system is based on a simple physics law; Static friction is greater than sliding friction. While the wheel rolls normally on the road, the friction that is applied between the two is the static one. However, when the wheel stops rolling (locks) while the vehicle is still in motion, the friction between the wheel and the road is the sliding one. As a result, by keeping the wheel rolling while braking, the forces that decelerate the vehicle can be greater than when the wheel slides on the road, resulting to a greater and safer braking with better control of the vehicle and shorter braking distances.

The construction of such a system relies on the turning frequency of each wheel; when the wheels have both the same frequency, there is no sliding, in contrast of when they do not, the wheel with the smallest frequency slides.

ABS can be found on the whole automobile industry, on every modern car or motorcycle, but quaintly not on bicycles, yet. There are some concepts (by the time of writing) from two companies (Bosch & BrakeForceOne) on such a product, however they are still on development and by no means ready to become consumer products. More information about them can be found in the links below:
http://ebike-mtb.com/en/bosch-ebike-abs-introduced/
http://www.brakeforceone.de/en/e-bike-abs/

From their approach the mechanical, part of the braking system (the caliper actuation) is made of hydraulic disc brakes, in contrast of this project's implementation which is made of mechanical disc brakes.

System Specifications

The system consists of two basic parts; the hardware/mechanical part and the software part.

Hardware Section

At this point it is essential to list all the used components. Starting from the mechanical part, the mountain bike that is used, is equipped with mechanical disc brakes whose discs have uniformly distributed holes for reasons that will be explained below.

Milestone 3 - Bicycle ABS

Disc Brakes

Furthermore, the hardware part is divided in two parts, the controller and its peripherals. More on the peripherals, they consist of two servomechanisms, two photointerrupter infrared sensors, two voltage regulators and one slider potentiometer and one 12 Volt battery. Reportedly to the controller's pcb, it consists of an Atmel ATMega328P microcontroller, one 16MHz crystal oscillator, one voltage regulator, 22pF capacitors and some resistors. Each component's purpose is discussed below.

Peripherals

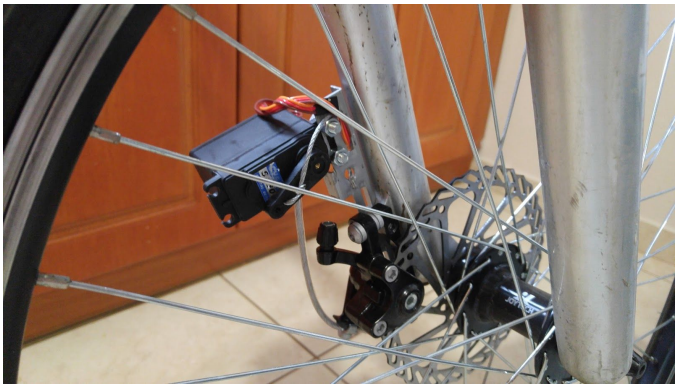
Servomechanisms

Connected to the disc brake's caliper lever as shown below, they serve the needed torque as instructed by the controller via a PWM signal.

Milestone 3 - Bicycle ABS

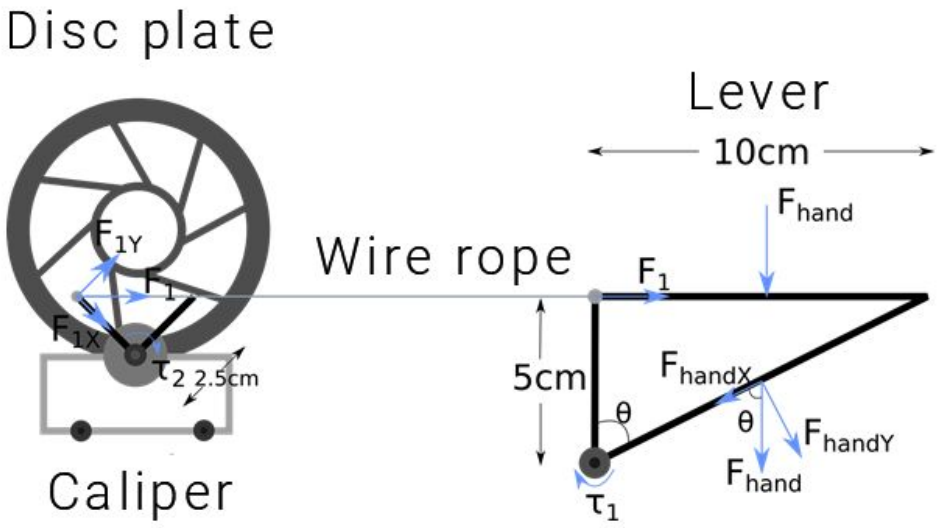
Servomechanism

Front Wheel



Rear Wheel

Physics Schematic



The suitable servomechanisms were selected after calculating the maximum needed torque on the disc brake's caliper lever. From physics, the needed torque can be calculated by transforming the torque applied on the hand lever. Let the above mechanical disc brake system and let F_{hand} be the force the rider's hand applies

on the lever. Let τ_2 be the torque applied from the wire rope to the caliper's lever, which is needed to be calculated. For the system above it is true that:

$$\tau_1 = F_{handY} * \frac{1}{2}\sqrt{5^2 + 10^2}\ kg * cm,\ where\ F_{handY} = F_{hand} * sin(\theta) = F_{hand} * \frac{10cm}{\sqrt{5^2+10^2}cm}\ kg$$

$$F_1 = \tau_1/5\ kg$$

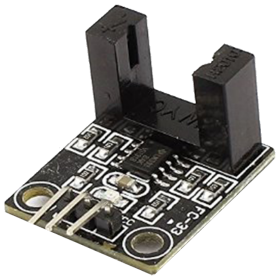
$$\tau_2 = F_{1Y} * 2.5\ kg * cm,\ where\ F_{1Y} = F_1 * sin(45^o)$$

So at last it is true that:

$$\tau_2 = 2.5 * F_{hand}\ \ kg * cm$$

By measuring the average maximum force applied from the rider to the lever, which is about 6 kg, and replacing it to the above equation, it can be calculated that the maximum needed torque τ_2=18 kg*cm. As a result, the selected servomechanisms can apply 30 kg*cm torque which ensures the correct and hard enough braking. On last but not least parameter is the turning time of the servomechanism which is needed to be less than 200ms per 60 degrees

Photo-Interrupter Infrared Sensors

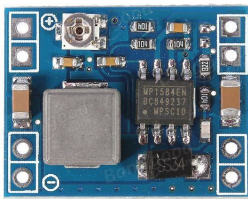


Positioned on the disc of each disc brake, they count the turning frequency of each wheel. Each sensor consists of one infrared LED and one infrared photoelectric sensor. They are both positioned interdimensionally. While the infrared ray between them is interrupted, the output is set. Using the disc brakes' discs holes as the infrared ray interrupting material, the sensor sets and clears its output while the wheel is turning. Because the disc's holes are uniformly distributed, the frequency can be measured without the knowledge of the wheel's phase by measuring the sensor's signal period.

Slider Potentiometer

Connected to the brakes' lever as shown below, it gives the controller the current braking level. Using it as a voltage divider, its position can be measured by comparing its output voltage to its input voltage.

Voltage Regulators



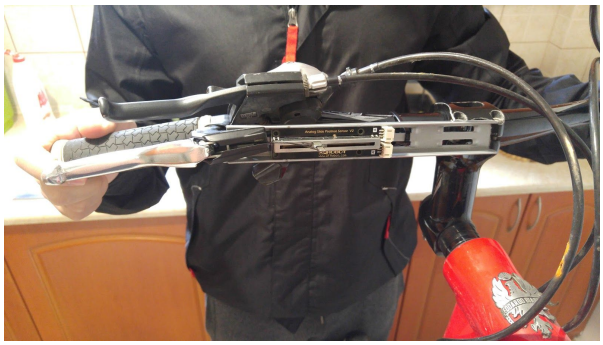
Regulating the battery voltage to 7.4 Volt for the servomechanisms.

Milestone 3 - Bicycle ABS

Mounting all the peripherals on the bicycle itself had to be done by constructing specific bases for each one, using metal bars that had to be cut down to match the needed size, some bolts and nuts and a lot of zip-ties.

Bicycle Side View

Slider Potentiometers



Front Wheel

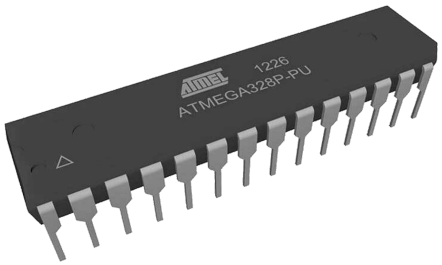
Rear Wheel

Under Seat

Milestone 3 - Bicycle ABS

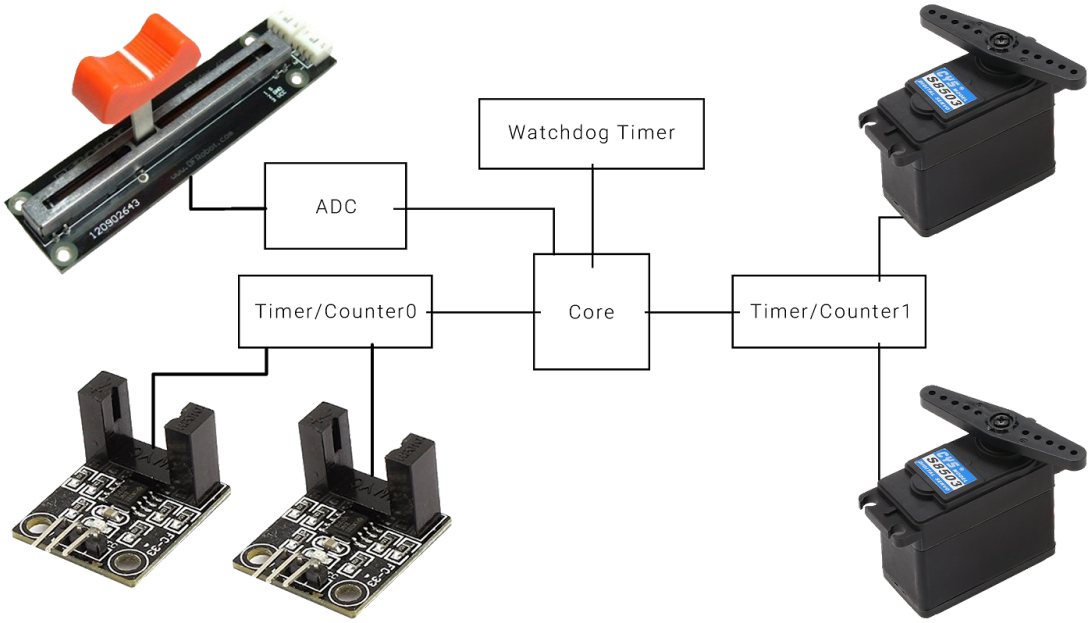
Controller

Atmel ATMega328P



All peripherals are connected to this 8-bit microcontroller as shown below.

Block Diagram



Starting from the left, the potentiometer's value is read by using the embedded Analog-to-Digital Converter (ADC), configured to auto-trigger conversions. The ADC divides the voltage range between 0 Volt and the reference voltage into 1024 parts and then it compares the input voltage with those parts to find the one with the minimum distance to return its value to the software which represents the position of the lever.

By using the Timer/Counter0 in CTC mode, an interrupt can be triggered every a specific frame of time. Using an Interrupt Service Routine (ISR) triggered by that Timer/Counter, a variable can be linearly increased on every interrupt handling so as to calculate the time. By connecting the Photo-Interrupters on an Interrupt Enable pin, as shown on the circuit diagram below, on every state change an ISR can be triggered; storing the current time on the rising edge, calculating the difference between the current time and the previously stored time to find the signal period on the falling edge. The essential accuracy is defined by the vehicle's (bicycle) maximum speed. Considering a maximum speed of 100km/h, it can be calculated that the maximum signal frequency is about 500 Hz as shown below.

The wheel's diameter is 2r=27" or 68.6 cm, so at each turn it goes through 2*π*r=215.4cm. When u = 100 km/h = 100000 m/h = 100000/3600 m/s = 27.77 m/s ≃ 27.8 m/s ⇒ u = 27.8 m/s then $F_{Wheel} = \frac{27.8\ m/s}{2.154\ m/turn} = 12.9\ turns/sec = 12.9\ Hz$. The disc of the disc brakes has 36 holes so the in every wheel turn there will be 36 interruptions of the infrared beam so 36 pulses on the sensor's output. As a result F_{signal} = F_{Wheel}*36 = 464.4 Hz ≃ 500 Hz.

At such speeds the signal's period is about T=1/500=2 ms. To be able to measure differences of 3 Hz at such speeds, T_{diff}=1/497 - 1/500 = 1.2*10^{-5}= 12 us. Setting the Timer/Counter0 to trigger an interrupt every 10 us is more than enough for this application, plus it considers the triggering time which is about 20 clock cycles on an empty ISR and also it leaves enough processing time for other calculations as 10 us stands for 160 clock cycles.

Each servomechanisms, as mentioned above, is controlled by a PWM signal whose duty cycle controls the servomechanisms angle. The PWM signal is generated by the Timer/Counter1 based on the servomechanism standard of 20 ms PWM period with minimum of 500 us of positive signal time and a maximum of 1440 us of positive signal time.

Lastly, a Watchdog Timer is used to avoid any unwanted behaviour due to electromagnetic noise or other reasons, setting it to reset the system every 16 ms if for any reason the proper program execution loses track, event highly impossible as the whole execution is interrupt driven.

16MHz crystal oscillator

Using it as an external oscillator, it serves the circuit the needed clock frequency with higher precision than the internal one.

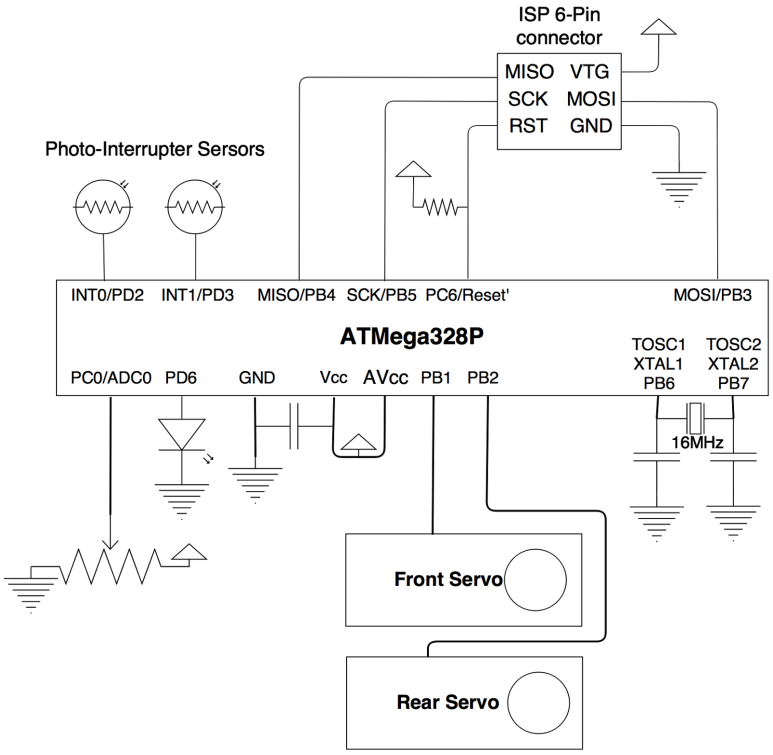
22pf Capacitors

Used to remove any parasitic noise from the circuit's DC signals such as the slider potentiometer output signal, power supply, input of oscillation, etc.

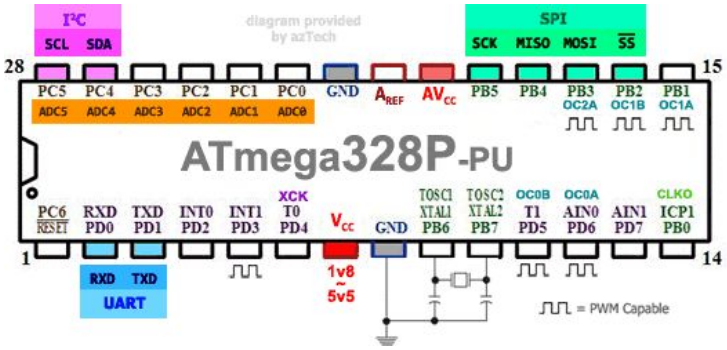
Resistors

Used to limit the current and to enable pull-up/pull-down on inputs.

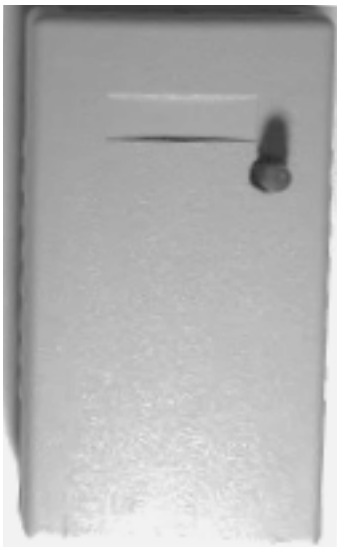
Circuit



ATmega328P Pinout



The circuit is kept safe from external factors in a custom made box, which is mounted using zip-ties under the riders seat as shown below.

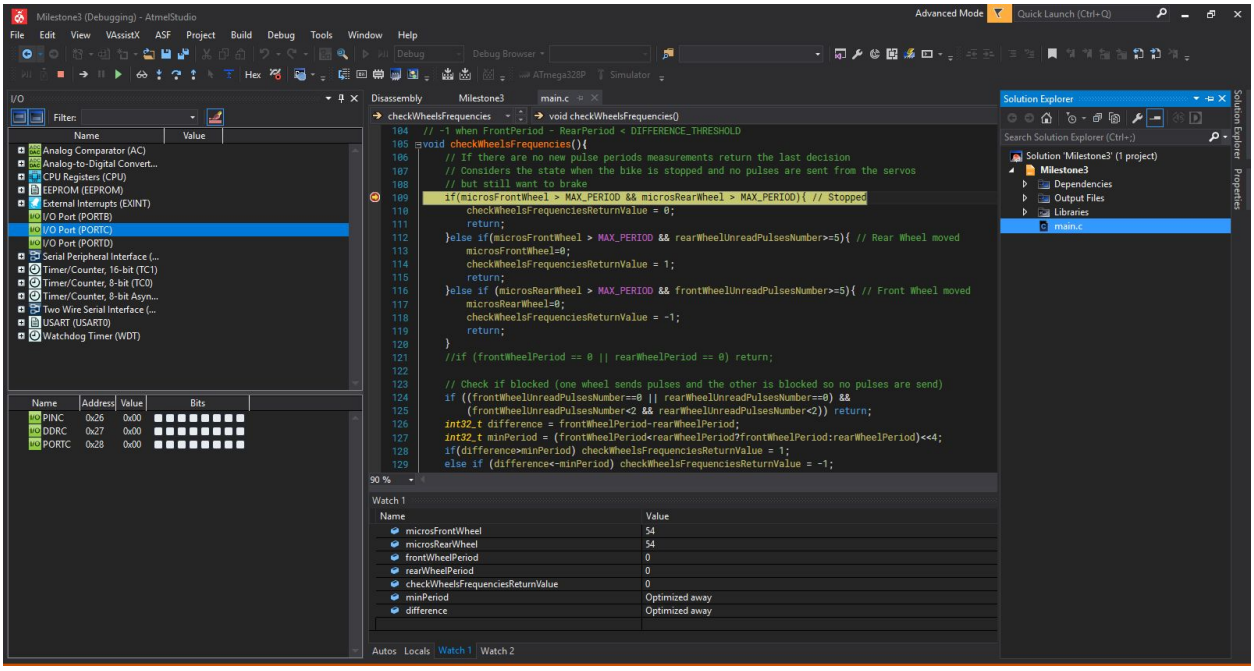


Software Section

Being a real time cyber physical system, it has to be time precise or else the consequences may be fatal. Consequently, the whole software development has to be done with the knowledge of each line's execution, so only the avr specific libraries were used which give the developer the ability to use all the configuration registers without using raw addresses but a set of defined (in the libraries) names of the addresses to make to code more readable, useable and maintainable. Additionally, hardware execution is preferred as it can compute in parallel to the main core other functionality such as the creation of PWM signals, creation of the time measurement, etc. Last but not least, interrupt driven execution is also preferred as it saves not only execution cycles but also energy compared to polling as well as it protects the execution track from being distracted from external factors such as electromagnetic noise, etc.

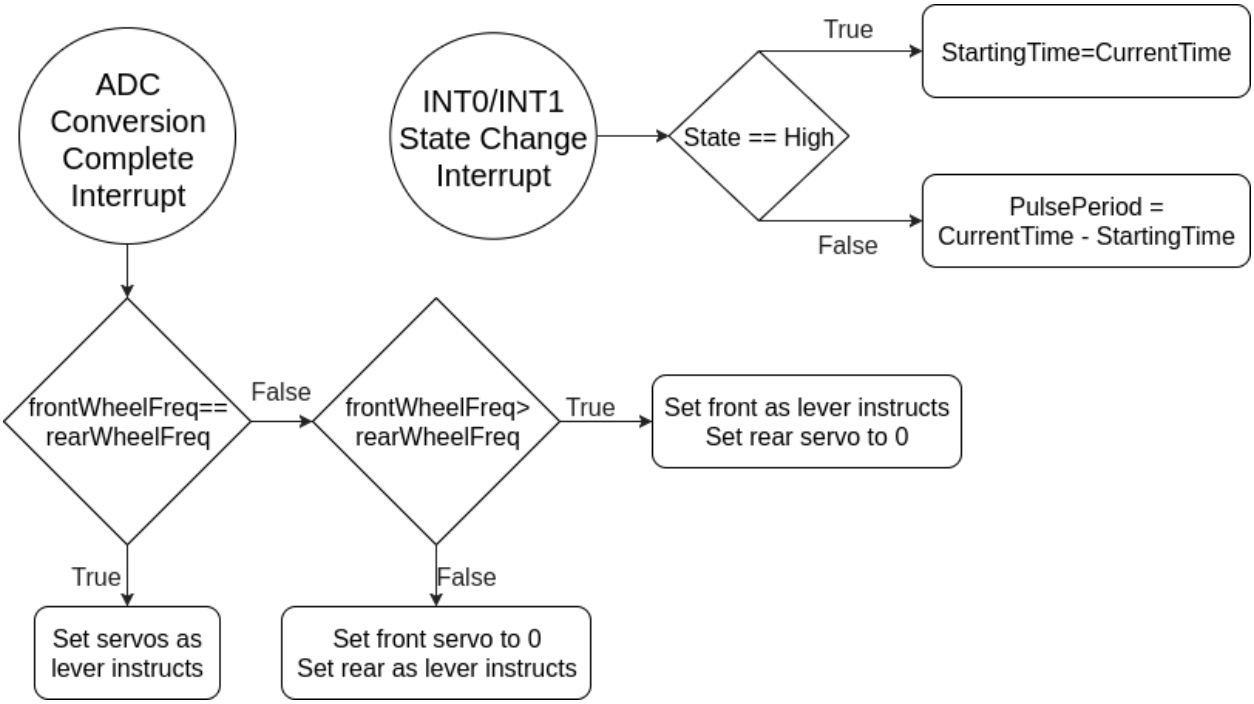
Although, the software was developed in Embedded C, however, C++ could also be used, in the Atmel Studio 7.0 IDE on Windows 10, the source code can also be compiled with avr-gcc compiler on any Linux Distribution. After the generation of the compiled .hex file which includes the software that can be flashed in the AVR's flash memory, it needs to be downloaded onto the AVR which was done using the ISP protocol via the AVRISP mkII. For ease of development, the ISP headers have been added on the circuit's main connector. The dowloading was done using the Atmel Studio 7.0 functionality, however, Avrdude can also be used for Linux Distributions.

Atmel Studio 7.0



The basic functionallity of the code can be seen on the flowchart below.

Flowchart



Future thoughts

Hydraulic Braking System

Since, if anything fails on the electronic system, the rider has to handle/stop a moving, possibly fast, bicycle with no brakes it is essential that the braking system is converter to a hydraulic one. Once the conversion is done, the ABS will work as an interrupter between the lever and the caliper through the brake oil, however, if the electronic part of the system fails, the brakes will still function as expected but wthout the ABS enabled.

Speed Meter - Distance Meter

Because of the already existance of the frequency meters (Photo-Interrupters sensors) on each wheel, its metering can be used to measure the current speed and also the distance that the rider has reached at any given moment. By adding to the system a display, statistics can be shown to the rider such as current speed, mean speed, trip distance, total distance, trip time, total time, current time, battery percentage, etc.

Automatic gearbox

Adding a frequency sensor on the bicycle's pedal (again the same use of the Photo-Iterrupter sensors) and servomechanisms connected to the front and rear derailleurs, the best gear ratio can be selected by comparing the pedal's frequency with the wheel's frequency.

Wireless Connections

As no one wants their bicycles bloated with wires, the system can be divided in tree subsystems; the front wheel (servomechanism, photo-interrupter sensor, battery, controller), the rear wheel (same as front wheel) and the lever (slider potentiometer, battery, controller). Each controller can sent and receive data / information about the state of each subsystem and decide its actions on its own, for instance, the front wheel subsystem broadcasts its current frequency, the rear wheel subsystem broadcasts its current frequency, the lever subsystem broadcasts its braking level and the front and back wheel controllers decide if they need to clamp or release the brakes by using the same algorithm as above.

GitHub

All source code and media files can be found on this project's GitHub Repository:
https://github.com/TFotakis/Bicycle-ABS