NXP Car

Design and implement a software for the NXP Allamak Car for autonomous driving on a track.

Studenti:

Olaru Mihai-Alexandru 1405B 

Axinte Silviu-Costin 1405B 

Barbir Dimitris - Tudor, 1405B

Project theme

This project aims to create an autonomous car that can run on a given track.

The main resource to monitor the track is the camera, whose output will be processed by the processor.

Necessary Hardware

For the chosen design we will use the following items:

- TSL1401 Camera
- BTN7960 Motor Driver
- FRDM-KL25Z Microcontroller

Resources management

Project

The project versioning will be dealt with in Bitbucket using Git.

https://bitbucket.org/tsm_ac/mainline/src/master/

Team

Every member of the team took one of the big parts of the project like this:

1) Olaru Mihai-Alexandru
    a) Setup Motor Control

b) PID implementation

2) Barbir Dimitris

a) Camera control

3) Axinte Silviu-Costin

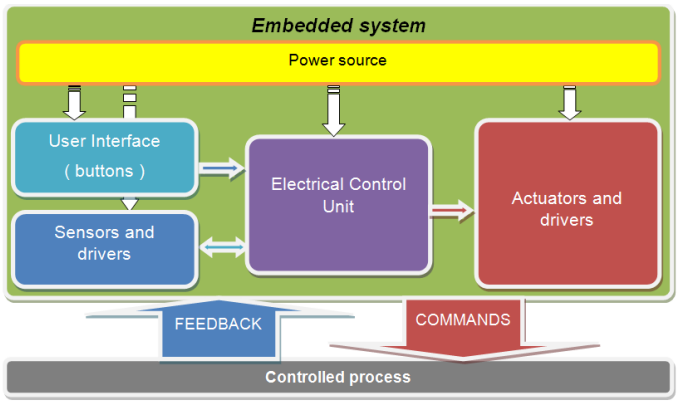
a) Integration of all the modules

Project functionality

Note: The track on which the car will run is white with black margins.

System design

Main components of an embedded system are shown in the following figure:



The camera

This camera will read a horizontally line of the track, and will return the values of each pixel (128 pixels).

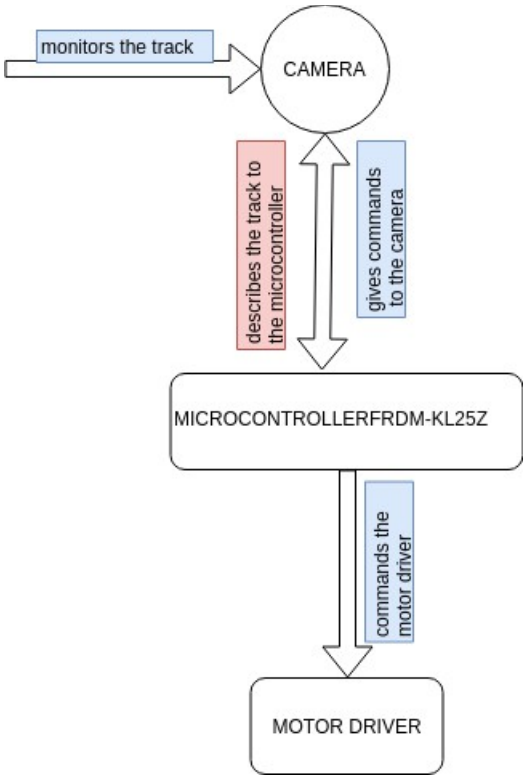
This camera is controller by the microcontroller. The microcontroller needs to set a clock, a signal to snap an image (in our case) of the track, and will receive an analogic input which represents the pixels values of the snapped image.

The microcontroller

The microcontroller has the responsability to control the overall functionality of the car. It reads the camera input and gives commands to the motors by the means of the motor driver.

The driver motor

It is controller by the FRDM-KL25Z microcontroller and it has the responsability to offer a good and stable control of the motors.



The control algorithm

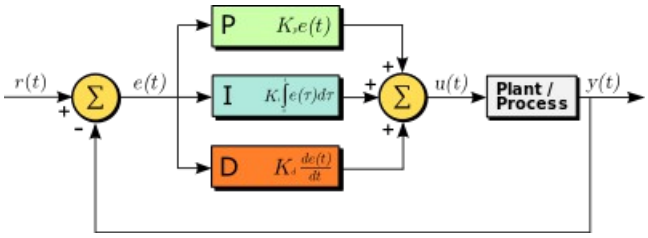
The main sensor that shall be used to control the car is the camera. The idea is to get the input from it as fast as possible, and find columns where the intensity of the pixels exceeds a threshold. In this way we are sure that we detect the outer lines.
After we detect the lines, we compute the center column (call it x value). This value is used to decide when to change the direction. We know that there are 128 pixels and if the car is on a right road the value of x shall be around 64. If at any time x is smaller than 64 then we need to make a turn to left.

To compute the steer value we use the PID algorithm. We have yet to find the best parameters for the formula, but we intend to run a simulation to help us.

To improve the steering, we also cotrol the two motors that power the back wheels. When the car decides to change the direction to left lets say, the motor on the left will slow down, this results in a cleaner way to take the curve.

The PID algorithm

A **proportional–integral–derivative controller** (**PID controller** or **three-term controller**) is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an *error value* e(t) as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted *F*, *I*, and *D* respectively), hence the name.

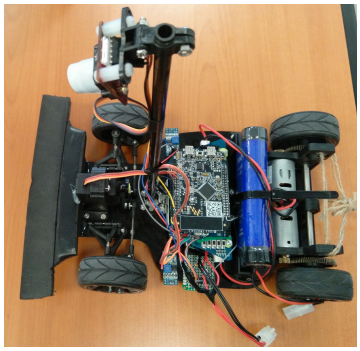


The overall control function can be expressed mathematically as

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt},$$

Where Kp, Ki, Kd all non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively (sometimes denoted *F*, *I*, and *D*).

Main sensor: Camera



Uses a linear sensor, TSL1401R, which consists of a single row of 128 photodetectors. What results is somewhat like peering through the narrow crack of a partially opened door to see a thin slice of what lies behind it. Some of the advantages of using this camera are the following:

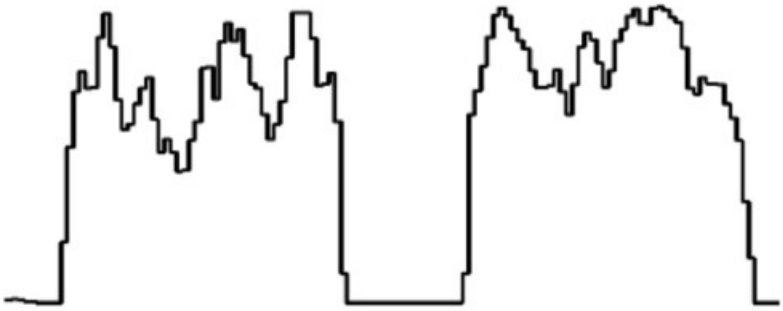
- Simple and easy to use
- Frequency of capture can be manipulated by the user
- Enough definition for line following application
- Removable and interchangeable lens for different resolutions

Besides the above benefits, the only disadvantage is the output signal of the camera is completely analog, which means the user has to be creative, to process this signal in order to make it “understandable”.

How is light interpreted?

As mentioned before, the camera is a combination of an image sensor (linear in this case) and a lens. The light that bounces from the environment enters through the lens, and the last one deflects light into the sensor. The sensor consists of a microscopic array of capacitors that gain charge depending on light intensity, therefore all pixel charge at the same time and the sensor releases each pixel value in one output signal one after the other until all pixel charges are released.

Because the camera is using a linear sensor it is impossible to gain a full view of the panorama in a single shot, therefore it only takes one line of the full panorama as shown in the next image. Here, the line to be captured is completely dependent on the distance to the lens as shown in the following figure.



To get signals from the camera, we need to take care of the following signals :

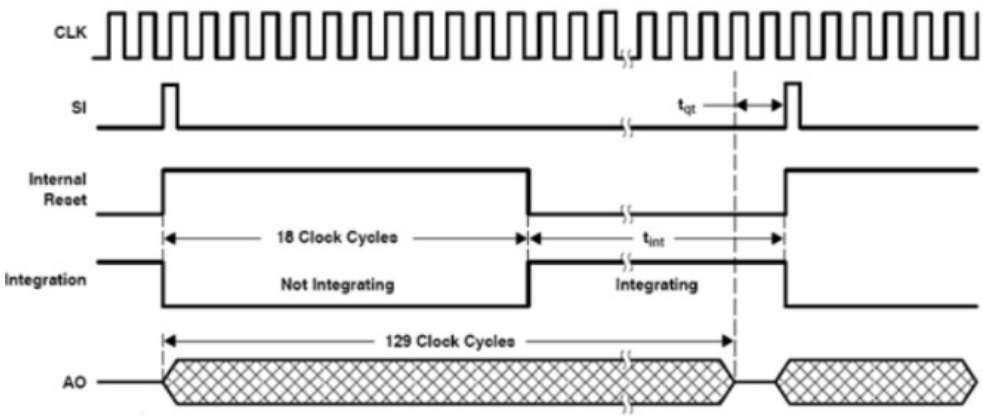
CK (clock), SI (serial input), AO (analog output); where CK and SI are camera inputs and AO is a camera output.

We can manipulate the frame speed of the camera by adjusting CK and SI signals. The faster is the CK frequency, the faster the camera releases the pixel values, and the closer each SI is from one another the faster each frame capture occur.

It is important to understand, the faster the frame capture occur the lower each pixel gain charge. This leads to another important factor, the “integration time”.

Integration time is the time the pixels have to complete its charge. With very long integration time, the pixels will be saturated even if there is low light intensity in the environment, on the other hand with very short integration time the pixel will not gain charge even if there is excessive light on the environment.

As it can be seen in the following figure, after 18 clock cycles, the pixels begin to charge back again. After 129 clock cycles all pixels are released from the camera, this means, from that moment it can send another SI pulse to release AO signal again; but as mentioned before longer the cycle, the pixels will charge more and user will get better pixel quality.


CLK
SI
t_{qt}
Internal Reset
18 Clock Cycles
t_{int}
Integration
Not Integrating
Integrating
129 Clock Cycles
AO