Teaching and development project for an eMobility bachelor course.

S.J.C. Koning
DC-Lab.org
THUAS
Delft, Netherlands
S.J.C.Koning@student.hhs.nl

T.D. Soons
DC-Lab.org
THUAS
Delft, Netherlands
T.D.Soons@student.hhs.nl

T.H.J. Ypma
DC-Lab.org
THUAS
Delft, Netherlands
T.H.J.Ypma@student.hhs.nl

E.D. Zatya
DC-Lab.org
THUAS
Delft, Netherlands
E.D.Zatya@student.hhs.nl

D.C. Zuidervliet
DC-Lab.org
THUAS
Delft, Netherlands
https://orcid.org/0000-0003-0833-5975

P.J. van Duijzen
DC-Lab.org
THUAS
Delft, Netherlands
https://orcid.org/0000-0001-5717-4333

Abstract—To teach eMobility to bachelor students, a development project was set-up, to be undertaken by students. The aim is to explore the various aspects of eMobility by practical laboratory tasks. In the first phase of the project, an eKart was assembled. In this phase of the project, the electronic controls are implemented. To finally implement drive-by-wire in an eKart, the speed controls, communication and the drivers-display, are implemented in a single steering wheel. Communication with the speed controller and the battery management system, has to be included in this steering wheel. By implementing all controls, communication as well as the drivers display inside a single steering wheel, simplifies the final eKart design. Communication via a single SPI interface reduced the wiring inside the eKart and allows easier exchange during maintenance or upgrades.

Index Terms—Smart steering, E-kart, Kart, Steering wheel, Motor drive

I. INTRODUCTION

By 2030, all new personal vehicles sold in the Netherlands will have to be zero-emission [1]. In order to keep up with the rapid movement towards electric vehicles and to educate students on the development of electric motor drivers, the DC-Lab [2] of the Hague University of Applied Sciences has given a group of bachelor electrical engineering students the task to design and build an electrical kart.

Within the DC-Lab an e-kart body with two 300W hub motors has already been made. To control these motors we need a master controller that is able to communicate to the slave motor drivers and a circuit board to take user inputs and show real-time feedback information. This PCB, printed circuit board, should be fitted inside a metal steering frame with a display. Besides the steering wheel, the motor driver needs a code update on the start sequence.

The motors are controlled by two Universal Four Legs (U4L’s) made by the DC-Lab [2]. These consist of four halfbrides that are combined using three of these to create a three phase AC inverter. This is needed since the hub motors are three phase motors. The TMS320F28069M µC from the C2000 series made by Texas Instruments is used with the motorware instaspin-FOC (instaspin-Field-Oriented Control) library made by Texas Instruments. This motor control library a speed control and a current control mode with a internal startup cycle. The complete hardware to control two or four wheel hub motors is shown in fig. 1.

![Fig. 1. Hardware to drive two or four wheel hub motors. Two three-phase inverters drive two independent wheel hub motors, but are controlled from a single C2000 microcontroller. The steering wheel connects via Arduino/SPI to the C2000/SPI interface.](image)

In this paper we dive into all parts concerning the design of this electrical kart. First we establish a system overview. Then we go into details on the design of the steering wheel, motor controller, the communication protocol between these two, the design of the kart itself, and finally we prove the validity our
II. SYSTEM OVERVIEW

In the system overview the overall system architecture of the kart is described, including the main subsystems and their relationship. A schematic overview is given in fig. 2.

There are three main parts that work together towards the end product of our system, i.e. making the motors turn with a desirable speed on our command. The E-kart consists of a steering wheel (yellow), reading the analog triggers and displaying information on a LCD screen. The motor controller (green), controlling two U4L’s and driving the motors. And finally the power supply (red), a battery with a fuse and circuit breaker protecting the motor controller from overcurrent.

For the driver to be able to control the speed of the motors the analog triggers need to be communicated to the C2000. This is done through a SPI bus. In turn, the C2000 sends back the speed and battery state for the Arduino to display on the LCD.

III. STEERING WHEEL

The purpose of implementing a smart steering wheel is to define user inputs, like acceleration and braking. Also, the smart steering wheel is a master controller for the C2000 driving the two motors. This master controller also displays speed and battery information on a LCD screen.

The steering wheel is split into two main parts, hardware, see fig. 4 and software. First the functionality of the arduino code is explained, afterwards the corresponding circuitry and PCB design is given.

A. Arduino

We use an arduino nano 33 IoT as the brain of the steering wheel. The main purpose of this micro-controller is to read the inputs given by the driver, convert them to something usable for the motor-controller, send them to the motor-controller, and finally receive the speed and battery status to display on a LCD screen. To visualise the code run by the arduino a block diagram is given in fig. 4.

The code is divided into three parts, initialisation and setup, the main script, and a 5 ms interrupt. In the initialisation the ADC and IO pins necessary for reading the input values are set, sending or receiving through the SPI bus is enabled and a 5ms interrupt is setup.

The interrupt routine consists of two parts, first two external functions are called which get the new input values from the gas and brake pedals, which are converted by the ADC, and buttons presses are checked. Secondly the converted brake or acceleration values are send to the C2000 through the SPI bus.

In the main script the values we get from interrupt routine are converted to something that the C2000 is able to use. Both the braking and acceleration values are converted to a 7 bit number. By sending an 8 bit value the eight bit can be used to indicate whether the motor-controller should brake or accelerate. Brake has priority over acceleration e.g. if both the brake and acceleration pedals are pressed in, the braking value will be send to the C2000 not the acceleration. How this works in practice will be expanded upon in chapter 5.E.

After the conversion is done, the button presses are processed. And the LCD is updated with the values received from the C2000. By counting 50 interrupts the LCD can update every 250 ms or 4 times per second.

B. LCD

The LCD is mounted in the top centre of the PCB on the steering wheel for a view of the kart’s speed and battery power. The speed is calculated from the diameter of the wheels and the frequency at which they spin. Battery power is measured on the basis of it’s voltage load, simply printing voltage. At a certain point the battery voltage is too low to supply enough power. If the voltage actually reaches 0 then the battery is inoperable. It is important to realise that the voltage supply decreases at a slower rate the lower the voltage becomes. As it is not generally known what voltage is 'low' the LCD will print “Battery Low!” at this point instead of it’s voltage load.

C. Circuit and PCB design

The main goal of the steering wheel PCB is to make it easier to connect the in and outputs of the arduino to the other parts in system.

To power the steering wheel PCB, 15V is coming from the C2000 through an STP-cable and is converted to 5V to comply
with the LCD screen. This conversion is done with a LM1117 5.0 [3]. The output of the LM1117 has an LED indicator to show whether the system is currently on or off.

All inputs from the steering wheel controls and the power lines are filtered to prevent noise. This is done by several RC combinations.

Solder pads can be used to route the inputs for the system directly to the C2000 instead of using SPI. This is done to make testing easier and gives a backup plan if the SPI communication fails to work properly.

From this circuit the PCB is drawn. The design rules regarding the PCB design are seen in table I. Power refers to the traces for earth, ground, 15V, 5V and 3.3V.

The PCB itself has 2 layers with a ground plane on both sides. Vias are placed connecting the bottom and top ground plane to prevent antennas from forming and causing disturbances in the signals. The RC filters are placed as close to the signals origin as possible with the trace going through the filter before being drawn to the destination.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Path width</th>
<th>Via size</th>
<th>Via drill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.2 mm</td>
<td>0.25 mm</td>
<td>0.8 mm</td>
</tr>
<tr>
<td>Power</td>
<td>0.2 mm</td>
<td>1 mm</td>
<td>1 mm</td>
</tr>
</tbody>
</table>

### IV. MOTOR CONTROLLER

#### A. Introduction

The 2 motors are controlled by a combination of a C2000 and two U4L’s. The U4L’s are setup as two three phase inverters that convert a DC battery voltage to a three phase AC. On these U4L’s are voltage and current senses that measure the three phase voltages and three phase currents to use these values for the motor control.

The C2000 µC is the controller that sends out the PWM signals based on the measurements using the motorware instaspin-FOC library from . This control consist of two different parts, namely the motor control part and the startup cyclus. First we explain the standard motor control, then the startup cyclus and we validate this design with simulations using the Caspoc software [4].

#### B. Motor driver

1) **Introduction:** The motor-ware instaspin FOC library generates PWM signals using space vector modulation using Clarke transformation. This modulation uses zero point shifting to amplify the currents with 1.15 compared with normal PWM generation. This library also bases the position of the motor on the three phase voltages and currents, transformed with Clarke transformation into $V_\alpha$, $V_\beta$, $I_\alpha$ and $I_\beta$. This is called sensorless field oriented motorcontrol[5]. This addition creates for cheaper manufacturing and repair options since the mechanical sensor that measures the position can be removed.

So a brief introduction of Synchronous Motors is given, then it is explained how the motorware instaspin FOC library works, then is explained how vectors can be used to control the motors efficiently.

2) **Synchronous Motors:** Synchronous motors work by applying a synchronous three phase voltage with a 120° difference. The maximum torque can be achieved if the stator and magnetic fields can be maintained at 90° phase shift.

This is where the FOC library comes in. The FOC controls the stator currents, represented as a vector, based on a projection that transforms a three phase time and speed dependent system into a two coordinate time invariant system. This two coordinate approach looks similar tot that of driving a DC motor.

In this time invariant system, maintaining the amplitude of the rotor flux at a fixed level leaves a linear relationship between the torque and the stator current. By controlling the stator current you can control the torque.
3) Separated Flux and Torque Control: One downside of the synchronous motor is that the torque and flux depends on each other. So with a control scheme such as the motorware instaspin FOC library can the flux and torque be separated from each other. This is done by making vectors of the flux and torque and setting two references: one of the flux and one of the torque. So by maintaining the flux at a fixed value is a linear relationship maintained between the torque reference and a variable torque component[5]. So if the currents are known, the motor can be controlled.

Three phase voltages and currents of the motor can be analyzed as three vectors with a 120° phase between the vectors. The stator current vector is a sum of the three current vectors with two spatial operators for the phase shift. Using Clarke transformation can three phase voltages and currents converted two vectors called \( \alpha \) and \( \beta \). The Park transformation takes the output of the Clarke transform with the variable flux to create a rotating reference frame where the flux is rotor flux position and the inputs from the Clarke transformation are now constants. Thus now are the flux component and torque component two components that can be controlled independently.

4) PWM generation: The output of the Park transformation is then transformed back with the added rotor flux position in mind as a new value with a inverse Park transformation to a Clarke transform. This is needed since Clarke transformation is used for the PWM generation for the U4L. In the Clarke transform a zero point shift, can be added to create increase in the current and voltage with the factor 1.15 outside the same bus voltage. From this transform are three PWM outputs created for each motor.

C. Simulations

In this case a kart must be pushed by the motors, which means there is a load on the motor for it to push through. If we apply this load the following happens in the simulations:

![Fig. 6. Full-Load input & output](image)

It is rather apparent that this of course does not just work. The motor does not have enough time to reach the same voltage level of the input signal, especially on the falling-edge. This also causes the current in the system to keep climbing infinitely, possibly creating dangerous situations.

V. Communication Protocol

A. Introduction

In order to make the kart drive, data exchange must take place between the steering wheel and the motor control. This is done by establishing a communication between the steering wheel (arduino) and the motor control(C2000), therefore a communication protocol has to be design. The 2 communication protocols that were considered are the Serial Peripheral Interface (SPI) or the Controller Area Network (CAN) bus. In this section, we will briefly discuss what these communication protocols are, what communication protocol is chosen for the kart and how it will be implemented.

B. Serial Peripheral Interface

Serial Peripheral Interface or SPI was developed to replace parallel interdace, this is done so it wont be necessary to route parallel bus around PCB. SPI protocol allowed a high-speed serial communication between a master, in our case the arduino, and a slave device, the c2000[6]. Furthermore SPI is much faster than Inter-Intergrated Circuit(I2C) and is currently the most common communication protocol between peripheral devices and IC’s[7].

One of the biggest advantage of SPI is that there no start and stop bits. This allow the data to stream continuously without any interruption[8]. Compare to I2C SPI has an uncomplicated save addressing system. As previously mentioned the SPI has a much higher transfer rate than other serial communication such as UART and I2C. Apart from the MOSI and MISO lines, 2 ways communication is possible, this is due to the fact that data can be sent and received at the same time. SPI has a few disadvantages such as:

- The communications must be well-defined in advance, you wont be able to send random data whenever you want.
- The controller must control all communications, peripherals can’t talk directly to each other.
- it requires separate CS lines to each peripheral.

C. Controller Area Network

Control Area Network or CAN bus is a low cost device network that is commonly use within vehicle communication. CAN bus was originally developed to replace the complex wiring with a two-wire bus. The specification requires transmit rates of up to 1 Mbps, virtually no concerns about electrical interference, and the ability to self-diagnose and fix data errors.
These have led to the popularity of CAN in the automotive, marine, medical, manufacturing and aerospace industries. CANBus allows the connected control units to communicate in a real time environment with limitations [9]. The CAN bus is not secured, and the behavior of connected devices can be manipulated and compromised [10]. For each device, the data in a frame is sent sequentially, it is done in such a way that if multiple devices are transmitting at the same time, the device with the highest priority takes precedence. Frames are received by all devices, including the sending device.

CAN protocol has its advantages over other communication protocol:

- Provide a very good price/performance ratio.
- Data transmission is very quick, up to 1Mb/s.
- The data is very reliable and has a very sophisticated and robust error detection [11].

But CAN protocol is not perfect, CAN protocol has its disadvantages too:

- Network must be wired in topology that limits stubs (a length of transmission line that is connected at one end only) as much as possible.
- High cost for software development and maintenance.
- Possibility of signal integrity issues.

D. Conclusion

Even though it is more common to use CANbus for vehicle we have decided to use the SPI. This is due to the fact that protocol is faster and much easier to implement. Furthermore this will make it easier for further development to 4-wheel drive kart.

E. Implementation

The Arduino (Master) read the value of the potential meter and button and send these value every 5ms to the C2000 (slave). The value of the potential meter correlate with the value of the acceleration and the brake. This value is then received by the C2000. To differentiate between acceleration and brake, we have established that integer value of 0 to 127 is brake and 128 to 255 is accelerate. This could be done due to the fact that the C2000 need IQ24 value and this goes from 0 to 127. The C2000 subsequently send the value of the speed of the kart and the SoC (State of Charge). How this value is measured is seen in Motor control code in appendix (insert measuring code).

VI. KART

To make the kart to be completely reproducible all physical components of the kart are documented. The framework of the kart was made by ROCmondriaan and therefore will not be elaborated upon. In this chapter all the added elements to the frame of the kart and the installation thereof is explained.

A. Battery management

To protect the system from overcurrent and potentially damaging current sensitive components a fuse is implemented between the system and the battery. Depending on the voltage of the battery a different fuse can be used with a different current rating to make the current protection as versatile as possible.

Another addition to the battery management system is a breaker switch. With this switch the driver can cut off power from the battery to the system instantly. This ensures work can be done safely without disconnecting the battery all together. While driving the kart the breaker switch also acts as an emergency shut off button if the system malfunctions.

B. Wiring diagram

The wiring diagram in fig. 1 shows us how all parts of the kart are connected. The steering wheel PCB is connected to the user inputs and the C2000 through STP cables with RJ45 connectors. This ensures the information gotten from the inputs and send to the motor controller has minimal noise. The motors are connected with banana and XT60 controllers to make sure the parts are easily interchangeable. The battery management system is connected to the motor controller with banana plugs as well. To find the correct inner diameters of the cables and their current ratings, the American wire gauge (AWG) standard is used [12].

VII. CONCLUSION

It is possible to make an electrical kart using easily available low costs electronic components.

By using the FOC instaspin library, it is possible to accurately drive 3 phase motors using sensorless control, without the need of a large amount of computing power. On top of that, the motor controller can be set up to enable regenerative braking and has the possibility for 4-wheel drive. The example coding delivered with the C2000 microcontroller, was enough to program the motor drive controllers.

The arduino nano 33 IoT was applied for the basic control functions inside the steering wheel. The arduino is configured as the master controller, able to direct multiple slave components.

Also the standard communication protocol which is used in the automotive industry, CAN, is not mandatory for communication in electric vehicles. By writing a communication
protocol between two micro-controllers using the SPI-bus we prove the versatility and functionality of the SPI.

All drivers control and display were integrated into a single steering wheel. This steering wheels communicates with the microcontrollers for the motor drive and with the battery management system.

REFERENCES