Documentation — Pin To Win —

Jürgen Markl

September 2021 - Dezembe 2021

Contents

List of Figures

List of Tables

Part I

µC Documentation

Chapter 1

Misc

1.1 Communication

1.1.1 USB Communication

Implementing a USB Device: Virtual Com-Port [1]

Speed: ca. 100 Chars → 7 µs

Attention: do not use ST-Link at the same time as USB Connection on µC → could damage USB Port of PC

Chapter 2

Solenoid driver

2.1 Duty Cycle Control

2.1.1 Draft with Python

Execution

For test purpose a python script was created with some generated user input to `button` to simulate button input. From this the output duty cycle `output` was calculated, with respect to the duty cycle. When the duty cycle was higher than the designed duty cycle, the output duty cycle was reduced.

Logic for duty cycle control The logic for controlling the duty cycle is implemented in the following code snippet:

```
# if solenoid should be activated
if button > 0:
        # if initial hit: full power
        if  prev_button != button:
                pass

        # if duty cycle exceeds limit
        elif duty > max_duty and output > max_duty:
                output -= 0.1

        # if below limit: set output to limit
        elif duty < max_duty and output < max_duty:
                output = max_duty
```

Duty Cycle is calculated either by a IIR approach or a FIR approach.

FIR-Filter For the FIR approach a time interval was defined over which the output duty cycle was added and a mean value was calculated. The result is shown in figure 2.1:

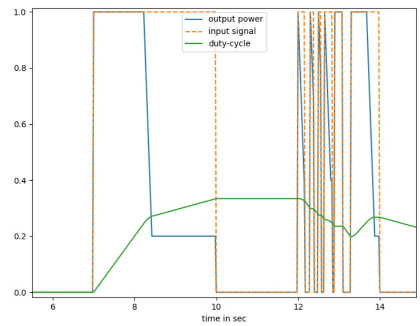

Figure 2.1: Duty cycle control with fir filter. Interval: 0.025s, calc_factor: 0.005

IIR-Filter For the IIR filter a filter coefficient was determined, which represents the weight of the current input with respect to the overall value (see figure 2.2)

Results and Conclusion

The IIR-filter is the easiest to calculate, because it does only require the value of the previous loop and 2 multiplications of floats. On the other had the FIR moving average calculation represents the previous time interval the best, because it holds all the history data. Comparing the two images, the FIR filter is clearly the most representative of the dissapated power, but also the most intensive in calculation. When a execution interval of 50ms is desired and the data will be accumulated over 5 seconds, with precision of 8 bit in storage, 100 byte of data will be required to be calculated as float to perserve precision, every loop. Further tests on the target microcontroller are required to determine the performance of the FIR-filter, to make a decision.

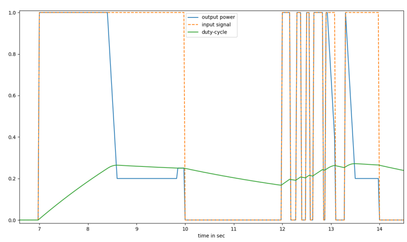

Figure 2.2: Duty cycle control with iir filter. Interval: 0.025s, calc_factor: 0.005

On the other hand, according to Application Note AN50003 of nexperia, you usually want a high current flowing until the shaft has traveled the whole length, after that the current is reduced to save power and prevent overheating. Following this approach, a total power record is not needed, as the solenoid is only driven a very short perild of time at max. current, depending on the time to reach its end point and then is reduced to a fraction of that current.

2.1.2 PWM generation

first experiments

Timer 2 Channel 2 to Pin A1 for PWM signal

Interval: 5 kHz

Calculate reload and prescaler value:

```
f_cpu = 96e6
f_timer = 5e3

f_cpu / f_timer = 19200 = reload_value
prescaler = 0
```

current values (02.09.21):

iir_const:	0.05
reload_value:	19200
output_reducer:	1920
loop_interval:	50ms

2.2 MOSFET Driver Circuit

For the Mosfet driver circuit a few were evaluated, depending on the availability of a mosfet driver IC, either this will be used and the schematics adapted accordingly, or a push-pull gate driver circuit will be used.

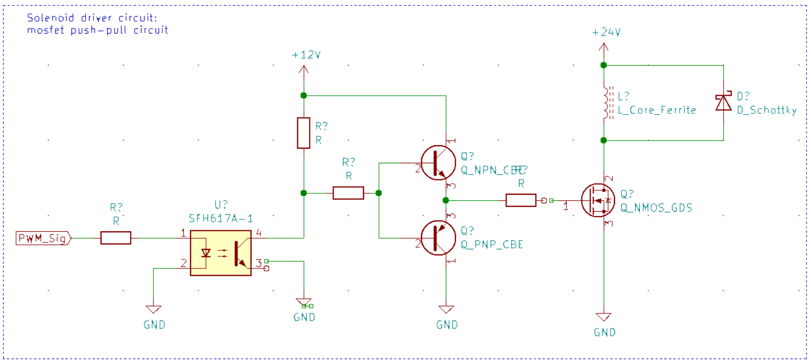

Figure 2.3: Solenoid driver circuit

For the evaluation of the different driver circuits, the Application note "MOSFET Gate Drive Circuits" [3] was consultated.

To prevent any noise interfering with analog signals used in the system, the mosfet and mosfet driver stage will be decoupled with a optocoupler from the µC circuit. A common ground will exist, but the ground at the power part will be routed separately and just join at a single point.

Requirements on the circuit:

- Current: 0.5A continuous, 2V at peak
- fast switching frequency of a few kHz
- reliable off-state

CON: for pwm pin: active low!

To comprimise this, the polarity of the PWM can be changed, but the solenoids will be activated, if not a main power switch is implemented (which is in general a good idea for falt detection)

Examples of some parts:

- MOSFET: DIODES Inc.: DMTH6016LK3

Chapter 3

Ball Detecton

3.1 Using Cameras

This would be the most straight forward approach, but was not chosen for the reasons of unique design and thinking outside the box

3.2 Touch Sensor

3.2.1 Magnetic and Capacitive Touch Sensors

Magnetic sensors would have been a good choice if the ball had some properties like a relative magnetic permeability that differed from 1 so it can be detected by creating a magnetic field below the the board.

For capacitive touch sensors the same principle applies. Detecting glass does not work with conventional touch sensors as used in smartphones, as well as the touch-sensor being expensive in a decent size.

3.2.2 Resistive Touch Sensors

Resistive touch sensors rely on the principle of two plates being pressed togehter and thus a different resistance can be measured across them when touched. The position of the touch can be calculated from the exposed contacts.

This method was discarded too, as we do not know at this stage, which ball we want to use and the max. weight of the ball would be 80g for the original ball, which for resistive touch sensors is fairly light.

3.3 Light Detection

A method to detect any object travelling over the surface was needed, as designs with a flat puck instead of a ball were evaluated. To achieve this, light sensors can be used to detect light passing through the board, or hitting the sensor sitting beneath the board with small holes to let light reach the sensor. If an object travels over the sensor, the change in light intensity is detected and can be evaluated to track the ball.

3.3.1 Evaluating Circuits

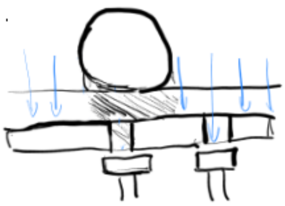

Figure 3.1: Light Sensor sitting beneath the board surface

Matrix Concept To track the movement of the object, many light sensor are required to be set up in a pattern around the area of interest. To precisely position the sensors beneath the board, a PCB can be manufactured, as the distances and positions of the sensors as well as the wiring will be easier and less messy.

To evaluate all sensors, where the sensor count can easily reach numbers beyond 100, a matrix formation can be used, where rows are evaluated individually and the sensors in one column are connected and only one column at a time is selected. To select the column, either a shift register, a BDC or a Shift Register can be used. To evaluate the rows, either a ADC with an input for each row can be used to receive an analog signal or a comparator with a fixed reference voltage can be used for digital representation of the data.

Sensors The choice of sensors is pretty straight forward, as cost is important and there should be a large amount of sensors, LDR (light dependent resistors) are the way to go: they are cheap and easy to evaluate. With the exponential decrease in resistance with a increased light intensity at the sensor surface, a threshold value can be determined fairly accurate. To evaluate the resistance, a simple voltage divider with a know resistor is enough. With the LDR being R1, the output voltage increases with an increase in light intensity.

Analog Evaluation The advantage of an analog evaluation of the data is, that more information is present to estimate the exact position of the ball at the time it influences more than 1 sensor. The disadvantage lies within the amount of data, that is generated and has to be processed and/or transmitted.

Digital Evaluation When evaluating the data only in binary format, the evaluation and scanning of the filed can be achieved much faster and transmitted in a more efficient way. The disadvantage of this approach is, that estimating the position is not as precise and the data rate is limited, as only two states are possible. When high precision is required, this approach is not advised.

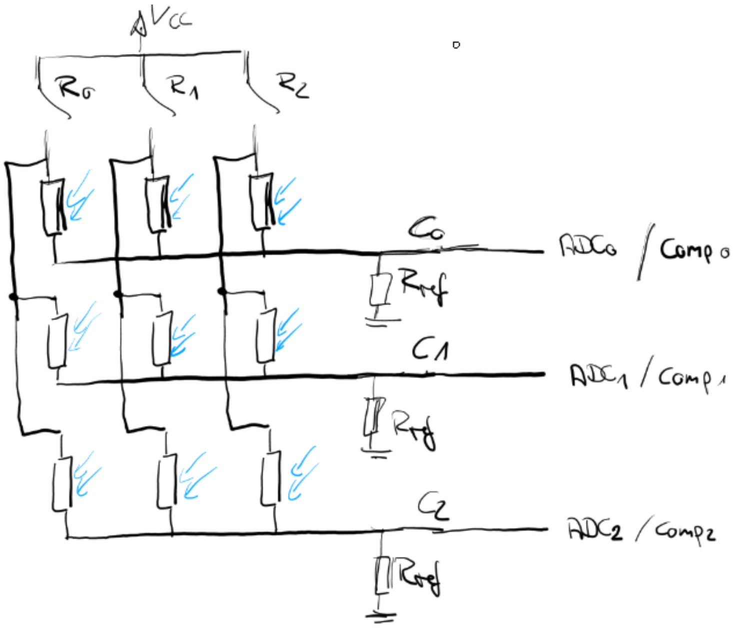

Figure 3.2: Matrix of LDR Sensor

Bibliography

[1] Stefan Frings: STM32F1 Anleitung
`stefanfrings.de/stm32/stm32f1.html`, last access: 02.09.2021

[2] nexperia: Application Note AN50003: Driving solenoids in automotive applications, Rev. 1.0, 4 December 2020

[3] Toshiba: Application Note - MOSFET Gate Drive Circuits, 26.07.2018