

Inhaltsverzeichnis

1. Motivation
2. Beratung und Bewerbung
3. Vorbereitungen
 - 3.1 Flug buchen
 - 3.2 Gesundheitliche Vorsorge
 - 3.3 Visum
 - 3.4 Wohnung
 - 3.5 Stipendium
 - 3.6 Gehalt und Lebenskosten
4. Arbeit und Sprache
 - 4.1 Präsentation des Instituts
 - 4.2 Das Praktikum
 - 4.3 Interkulturelle Arbeitsweise
 - 4.4 Die Sprache
5. Reisen und Ausflüge
 - 5.1 Peking
 - 5.2 Shanghai
 - 5.3 Xi'an
 - 5.4 Umgebung von Shanghai
 - 5.5 Hongkong
6. Kultur
 - 6.1 Politische Einstellung
 - 6.2 Mentalität und Verhaltensweisen

1. Motivation

Ich habe mich aus verschiedenen Gründen für ein Praxissemester in Shanghai entschieden. Ein Grund war, dass ich die chinesische Sprache und Kultur näher kennenlernen wollte. Außerdem erhoffte ich mir, durch die Zusammenarbeit mit Studenten und Dozenten, die chinesischen Arbeitsweisen kennenzulernen, damit ich diese in mein späteres Berufsleben erfolgreich einbringen kann.

Die Entscheidung mein Auslandssemester in China zu absolvieren fiel recht schnell. Chinas Wirtschaft wächst mit atemberaubendem Tempo. In den nächsten Jahren wird China eine noch entscheidende Rolle in der Weltwirtschaft spielen und jegliche Erfahrungen mit diesem Land können für mich hilfreich sein. Außerdem war ich noch nie in China und es gibt viele faszinierende Orte die ich mir gerne ansehen möchte.

2. Beratung und Bewerbung

Beraten wurde ich von Prof. Dr.-Ing. Fritz Pörnbacher. Dieser informierte mich zum ersten Mal während einer Vorlesung über die Möglichkeiten eines Praxissemesters in Shanghai und über die ins Leben gerufene Kooperation mit dem Shanghai Technical Institute of Engineering & Information (STIEI).

Ich ließ Prof. Dr.-Ing. Fritz Pörnbacher meine Bewerbung wenige Tage später zukommen. Obwohl zu Beginn nur für zwei Personen eine Praktikumsstelle in Shanghai zur Verfügung stand, sicherten die chinesischen Partner allen sechs deutschen Studenten eine Anstellung an dem STIEI zu.

Alle Studenten studieren im gleichen Jahr Elektro – und Informationstechnik. Wir haben uns bei der Anmeldung gegenseitig unterstützt und uns die Arbeit aufgeteilt. Dies hat uns viel Zeit erspart und wir haben uns viel geholfen.

3. Vorbereitungen und Anmeldung

Trotz Internet sollte die große Entfernung mit Zeitverschiebung nach China nicht unterschätzen werden. Viele kleine Probleme und Fragen lassen sich per E-Mail nur schwer klären, obwohl sie mit einem kurzen persönlichen Gespräch leicht aus der Welt zu schaffen wären.

3.1 Flug buchen

Wir haben uns schon früh um einen Flug gekümmert. Gebucht haben wir Mitte Mai. Schwierig ist, dass aus Deutschland nicht viele Direktflüge nach China gehen. Deshalb haben wir uns dafür entschieden, dass wir eine drei tägigen Aufenthalt in Dubai machen. Es ist zu empfehlen sich früh zu kümmern, damit der Flug nicht zu teuer wird. Den Rückflug haben wir von Hongkong aus gebucht, damit wir uns auch diese Metropole noch am Ende unseres Praxissemesters ansehen können. Als Anreise reichte den chinesischen Partnern die erste September Woche, somit waren wir mit unserer Anreise sehr flexibel.

3.2 Gesundheitliche Vorsorge

Vor Abflug sollte man unbedingt noch einen Arzt aufsuchen, damit veraltete Impfungen aufgefrischt und benötigte Impfstoffe für China verabreicht werden können. Abgesehen von Hepatitis A und Hepatitis B sind normalerweise keine besonderen Impfungen notwendig.

Außerdem muss eine Auslandskrankenversicherung für nicht EU Länder abgeschlossen werden, damit auch im Ausland ein Versicherungsschutz besteht. Ein gutes und preiswertes Angebot stellt dabei die MLP Finanzdienstleistungen AG mit Sitz in Landshut.

3.3 Visum

Der Bewerbungsprozess ist sehr genau auf der chinesischen Botschaftshomepage beschrieben. Hier erhält man auch alle Formulare und Dokumente, die man an dem Tag des Interviews mit zur Botschaft bringen muss. Diese bestehen aus vorgedruckten Formularen und selbst ausgewählten Dokumenten. Das Visum nimmt sehr viel Zeit in Anspruch, außerdem müssen alle Dokumente korrekt und ausführlich dargelegt werden, sonst wird das Visum verweigert. Auch wir müssten zweimal zur Botschaft nach München fahren, weil es beim ersten Mal Probleme mit der Einladung der Chinesen gegeben hatte und wir somit kein Visum bekommen haben. Nach einigen Tagen konnten wir dann unseren Reisepass mit eingeklebten Visum abholen.

3.4 Wohnung

Die Wohnungen befanden sich in Hongmiao und wurden uns kostenlos zur Verfügung gestellt und deshalb auch bereits Monate vor unsere Ankunft von den chinesischen Partnern ausgesucht. Auch die monatlich anfallenden Nebenkosten durch Strom, Wasser und Heizung wurden von der Partnerhochschule bezahlt. Insgesamt bekamen wir zwei Wohnungen, also in jeder Wohnung drei Studenten. Jedoch war die Wohnung nur für zwei Personen ausgelegt, sodass die dritte Person auf einem harten Holzgestell hätte schlafen sollen, doch in beiden Wohnungen wurde auch dafür eine passende Lösung gefunden. Nach einigen Reparaturen waren die Wohnungen dennoch zu unserer Zufriedenheit.

Direkt vor unseren Wohnungen befand sich ein Markt, auf dem es jeden Tag frisches Obst, Gemüse und Fleisch gab. In oftmals weniger als fünf Gehminuten waren einige Restaurants und Supermärkte erreichbar auch die Arbeit war nach zehn Minuten erreicht.

3.5 Stipendium

Wir vereinbarten einen Termin mit dem International Office der HAW Landshut und wollten uns über Möglichkeiten für einen finanziellen Zuschuss erkundigen. Am Ende bekam jeder Student von dem Deutschen Akademischen Austausch Dienst (DAAD) einen Zuschuss von 500,-€. Eine genaue Frist gibt es dabei nicht, jedoch empfiehlt es sich möglichst schnell den nötigen Antrag abzugeben.

3.6 Gehalt und Lebenskosten

Das Gehalt beträgt nach Arbeitsvertrag 500,- € pro Monat, welcher Betrag das in Yuan (chinesische Währung) ist hängt ganz vom Wechselkurs ab. Es war uns auch freigestellt, ob wir das Gehalt in Euro oder Yuan bekommen. Wir wählten bis auch unser letztes Gehalt immer Yuan, weil Geldwechseln in China nur in der Bank möglich ist und dies immer viel Wartezeit mit sich bringt.

Wie viele Kosten man hat, hängt stark davon ab wie viel Geld man ausgeben möchte. Kosten für Flug und Visum wird man fest einplanen müssen, aber Leben kann man in China auch sehr günstig. Besonders in Hongmiao (Fengxian), das ein süd-östlich gelegener Stadtteil von Shanghai ist, wo die verschiedensten Gerichte bei den unzähligen Straßenimbissen oft weniger als 0,50 € kosten und ein Friseurbesuch mit 1,50 € auch sehr günstig ist.

Jedoch empfiehlt es sich sehr jedes Wochenende direkt nach Shanghai zu fahren und auch die umliegenden Städte wie Nanjing, Hangzhou, Wuxi oder Suzhou zu besuchen. Ebenso reisten wir mit dem Schnellzug nach Peking und mit dem Nachtzug nach Xian.

Trotz der monatlichen Vergütung von 500,-€ sollte vor dem Abflug ein Budget von ca. 2.000 € zur Verfügung stehen, damit keine finanziellen Probleme im Ausland auftreten.

4. Arbeit und Sprache

4.1 Präsentation des Instituts

Das Shanghai Technical Institute of Engineering & Information (STIEI) wurde 1959 von zehn Hao gegründet und heute besuchen etwa 8000 Schüler und 430 Lehrer die Schule. Das STIEI gehört zu den 100 nationalen Eliteberufskollegs, die besonders gefördert werden. Es ist auch die einzige Berufsschule in Shanghai, an der die Hanns-Seidel-Stiftung in Kooperation mit dem Erziehungsamt eine duale technische Berufsausbildung durchführt. Seit das Berufsbildungszentrum 1985 gegründet wurde, werden hier dreijährige Ausbildungen zum IT Systemelektroniker, zum Elektroniker für Betriebstechnik, zum Mechatroniker und zum Mechatroniker für Werkzeugmaschinen durchgeführt. Das Institut ist in die Abteilungen, Elektrotechnik, Kommunikation & Information, IT, Maschinenbau, Wirtschaft & Management und Animation, unterteilt. Unter dem Dach des STIEI gibt es heute neben dem Berufskolleg mit den sechs Fakultäten noch sogenannte „Sekundär-Akademien“, wie das Berufsbildungszentrum Shanghai (BBZ) und die chinesisch deutsche Berufshochschule (CDBH). Das BBZ ist die einzige Schule in Shanghai, die von den chinesischen Bildungsbehörden und der Hanns-Seidel-Stiftung in Kooperation betrieben wird. Die STIEI besitzt eine Ausbildungskooperation mit der Technikerschule Passau sowie der Hochschule Landshut. Eine Besonderheit der chinesisch deutschen Berufshochschule (CDBH) ist, dass die Studenten, welche die Ausbildung in China erfolgreich abschließen, anschließend ein Studium an der Hochschule Landshut in den entsprechenden Fakultäten beginnen können.

4.2 Das Praktikum

Unser Praktikum fand in der chinesisch deutschen Berufshochschule (CDBH) statt. Unser Praktikum begann eher durchwachsen, obwohl wir einen Dolmetscher hatten, war die Kommunikation mit unseren beiden Verantwortlichen Herrn Li und Herrn Luo nicht immer einfach. In der zweiten Arbeitswoche wurde, nach anfänglichen Unklarheiten, unser weiteres Vorgehen besprochen und festgelegt (siehe Projektplan). In den darauffolgenden Wochen wurden wir oftmals unnötig aufgehalten, weil die Besorgung einfache Geräte wie Oszilloskop, Multimeter oder Netzgerät in China mehrere Tage dauern kann. Dennoch nahm unser Projekt Schritt für Schritt Gestalt an, sodass wir am Ende den Verantwortlichen der CDBH ein brauchbares Projektergebnis vorzeigen konnten. Wir 6 Studenten wurden auf 3 Abteilungen aufgeteilt. Kommunikations-, Informationstechnik – und Elektrotechnik. Mein Partner und ich waren in der Elektrotechnikabteilung und unser Projekt beinhaltete ein Line-Tracking-Auto (Auto, das mithilfe von Sensoren eine schwarze Linie erkennt und dieser auf einem Parkour folgt).

4.3 Interkulturelle Arbeitsweise

Unsere interkulturellen Kompetenzen wurden in China stets gefordert. Die Verantwortlichen des Projekts hatten zu Beginn des Praktikums Anforderungen, welche in 4 Monaten nicht umzusetzen gewesen wären, deshalb musste in einem gemeinsamen Gespräch eine Lösung gefunden werden, welche für alle zufriedenstellend war. Auch jegliche Aussagen von Chinesen mussten wir stets sehr kritisch betrachten, weil diese ihre Unwissenheit niemals zugeben und deshalb eher falsche Aussagen treffen, bevor Sie ihr Nichtwissen zum Ausdruck bringen würden. Unsere oftmals dringend benötigten Bauteile oder Geräte wurden häufig erst nach mehrmaliger Nachfrage besorgt, weil sie sonst unsere Anfragen als nicht wichtig betrachten.

4.4 Die Sprache

Die Verständigung mit den chinesischen Professoren erfolgt entweder über Englisch oder es wird ein Dolmetscher dazu geholt. In der Arbeit kann man sich einigen Studenten und jüngeren Professoren auf Englisch unterhalten. Jedoch empfiehlt es sich, wenigstens ein paar einfache Floskeln auf Chinesisch zu lernen, weil in den meisten Teilen Chinas kein Englisch gesprochen wird. Die chinesische Sprache ist sehr schwer zu lernen, deswegen ist es umso besser, je früher man mit dem Chinesisch lernen anfängt. Es ist wichtig mit chinesischen Studenten einen Kontakt aufzubauen, diese haben uns oft über verschiedenen Veranstaltung an der Schule informiert.

An dem Shanghai Technical Institute of Engineering & Information arbeiten ein paar Dolmetscher, diese sprechen sehr gut Deutsch und haben uns bei allen Anliegen schnellstmöglich geholfen.

5. Reisen und Ausflüge

5.1 Peking

Nach Peking fuhren wir mit einem Schnellzug (fährt konstant mit 300 km/h). Es dauerte ca. 4 Stunden und es kostete umgerechnet ca. 140 €. Ich empfehle, wie wir, einen mindestens 5-6 tägigen Besuch, da es sehr viele Sehenswürdigkeiten zu sehen gibt. Dies ist ideal für die erste Oktoberwoche (Herbstfest), in welcher in China eine komplette Woche frei ist.

5.2 Shanghai

Wenn wir nicht unbedingt gezwungen waren, aus Arbeitsgründen in Hongmiao zu bleiben, fuhren wir eigentlich immer nach Shanghai-City. Es gibt dort so viel zu sehen und jeder Einzelne von uns hat diese Stadt geliebt. Das Essen, die Sights und auch das Nachtleben sind sehr gut. Allerdings ist es natürlich teurer als am Land wie z.B. Hongmiao.

5.3 Xi'an

Eine Reise nach Xi'an ist sehr zu empfehlen, da deren Sehenswürdigkeiten zu den wichtigsten Chinas zählt (Terrakotta-Armee). Allerdings fährt man mit einem Schlafzug ca. 14 h einfach. Hin- und Rückpreis des Zugs beträgt ca. 75 €. Wir haben wegen dieser Reise nach einem freien Tag am Freitag bei der Schulleitung gebeten, den wir natürlich dann in den Mittagspausen nacharbeiteten. Effektiv hatten wir dann 2 Tage zum besichtigen.

5.4 Umgebung von Shanghai

5.4.1 Suzhou und Wuxi

Suzhou und Wuxi liegen direkt nebeneinander. Wir fuhren am Freitag mit dem Zug nach Suzhou, sahen uns am Samstag Suzhou an und abends ging es weiter nach Wuxi mit dem Zug. Von Shanghai nach Suzhou dauerte es ca. 30 Minuten und von Suzhou nach Wuxi ca. 15 Minuten. Insgesamt kosteten die Tickets ca. 20 €. Suzhou ist das „Venedig Chinas“ und ist auch auf alle Fälle eine Reise wert. In Wuxi gibt es den Lingshan Buddha zu besichtigen, welcher 88 Meter hoch ist und einfach nur faszinierend ist. Alles in allem auf einem Wochenende machbar.

5.4.2 Hangzhou

Hangzhou liegt im Südwesten Shanghais und ist mit dem Schnellzug ca. 1 Stunde entfernt. Die Stadt Hangzhou besticht durch seinem gigantischen See, dem „West Lake“. Auch hier gibt es eine Menge an Attraktionen zu besichtigen. Allerdings hatten wir Pech, da an diesem Wochenende der Smog-Nebel in China sehr extrem war und wir dadurch nicht den besten Ausblick hatten. Zugticket Hin- und Zurück kostete ca. 20 €.

5.4.3 Nanjing

Nanjing liegt 270 km im Westen von Shanghai. Nanjing war einmal die Hauptstadt Chinas und ist sehr sehenswert. Im 2. Weltkrieg gingen hier Massenvergewaltigungen der Chinesen von Japaner zugange, welche einen deutlichen Eindruck auf Denkmäler und Sehenswürdigkeiten der Stadt hinterlassen hat. Außerdem war Nanjing Mitaustragungsort der Olympiade 2008. Das Olympiagelände wird nach wie vor für sportliche Aktivitäten verwendet und besticht in seiner Größe und modernen Bauweise. Nanjing hat auch einen großen See, den „Xuanwu Lake“, an den ein Berg namens „Zhongshan“ angrenzt. In dieser „Zhongshan Mountain Scenic Area“ gibt es noch drei wichtige Sehenswürdigkeiten: „Xiaoling Mausoleum“, „Dr. Sun Yat-sen Mausoleum“ und dem „Linggu Tower“. Alles in allem einen Wochenend-Trip auf alle Fälle wert.

6. Kultur

Die chinesische Kultur ist eine sehr spezielle und alte Kultur, wobei viele Verhaltensweisen beachtet werden müssen. Jedoch wird den Ausländern eine wegen Unwissenheit unabsichtliche Missachtung der chinesischen Verhaltensweisen schnell verziehen und Sie verstehen durchaus, dass in den westlichen Ländern diese Verhaltensweisen nicht so verbreitet sind. Grundsätzlich sollte man den Chinesen immer freundlich gegenüber treten, dann können kaum Fehler gemacht werden.

6.1 Politische Einstellung

In der VR China herrscht eigentlich ein Einparteiensystem und auf allen Ebenen der Macht gibt es sehr enge Verbindungen zwischen dem Staatswesen und der Partei. Unabhängig von ihrer politischen Meinung sind nahezu alle Chinesen sehr stolz, dass ihr Land heute zu den mächtigsten der Erde gehört. Durch einige Gespräche mit jüngeren Chinesen ist uns deutlich geworden, dass diese sehr wohl über die eingeschränkte Freiheit in Ihrem Land Bescheid wissen und unzufrieden damit sind, jedoch würden Sie Ihre Unmut niemals öffentlich zum Ausdruck bringen. Die Angst vor den Folgen schrecken alle Chinesen ab. Heikle Themen, welche man als Ausländer in China lieber nicht ansprechen sollte sind die Unabhängigkeit von Tibet und Taiwan oder der Machtanspruch der Partei. Auch Korruption, Umweltverschmutzung und Justizentscheidungen werden in privaten Runden oft diskutiert, jedoch sollte auch von diesen Themen Abstand gehalten werden.

6.2 Mentalität und Verhaltensweisen

Jedes Geschenk wird mit China mit beiden Händen überreicht, mit einer leichten Beugung des Oberkörpers entgegen genommen. Das Überreichen der Karte mit einer einzigen Hand gilt als Geringschätzung des Gegenübers.

Beim Essen in China bestellt nie jemand für sich allein eine Speise, sondern es wird eine Reihe verschiedener Speisen bestellt und alle probieren von allen Speisen. Ein gutes Essen mit Chinesen sollte sich durch die Vielzahl verschiedener Speisen auszeichnen und von einem guten Essen sollte immer etwas übrig bleiben (sonst ist es beschämend zu wenig bestellt zu haben)

Genussvolles Schlürfen und lautes Schmatzen sind nicht unhöflich und gelten in als Ausdruck der Wertschätzung für ein gutes Essen. Lautes Niesen und Schnäuzen ins Taschentuch gelten (nicht nur beim Essen) hingegen als abstoßend. Auch Trinkgeld gilt in China als ungewöhnlich und befremdend. Es gibt noch unzählige weitere Verhaltensweisen, darüber Bescheid zu wissen ist bestimmt von Vorteil und auch zu empfehlen, dass man vor Abflug sich dahingehend informiert.

Shanghai Technical Institute of Electronics and Information

Department of Computer Science

Project Documentation

**Topic:
"The Seesaw Balance Robot"**

Project Manager:
Dong Changchun
Dr. Fritz Pörnbacher

Students:
Felix Wutzer
Fabian Bernwinkler

Period:
09. September 2013 – 27. Dezember 2013

Table of contents

I. Cover Sheet

II. Table of Contents

1. Definition	1
1. a) Task	1
1. b) Specific Test Sequences	2
2. Angle Measurement	3
2. a) Sensors	3
2. b) Complementary Filter	9
3. Electrical Car	11
3. a) Manual	11
3. b) Chassis	14
3. c) Motor Driver	15
4. Line Tracking	17
4. a) Sensors	17
4. b) Procedure	18
5. Micro Controller	19
5. a) Controller Introduction	19
5. b) Configuration	20
6. Control Algorithm	26
7. Results	27
7. a) Functionality	27
7. b) What can this project be used for?	28
8. Code	29
8. a) global_variables.h	29
8. b) Header.h	30
8. c) L3GD20_registers.h	32
8. d) Main.c	33
8. e) uC_config.c	35
8. f) Sensor_Init.c	38
8. g) SPI.c	39
8. h) PCA_ISR.c	42
8. I) T0_ISR.c	45
8. j) Control.c	46
8. k) Acc_Calc.c	48
8. l) steer.c	49
8. m) Sequences.c	50
8. n) Controller_Output.c	54

a) Task

The goal of this project is to design and manufacture an electric car that balances on a mechanical seesaw. If a counterweight is placed on the seesaw, the car has to change its position to achieve a plane alignment. The car has to react automatically to every change of position of the counterweight and keep the balance for a certain amount of time.

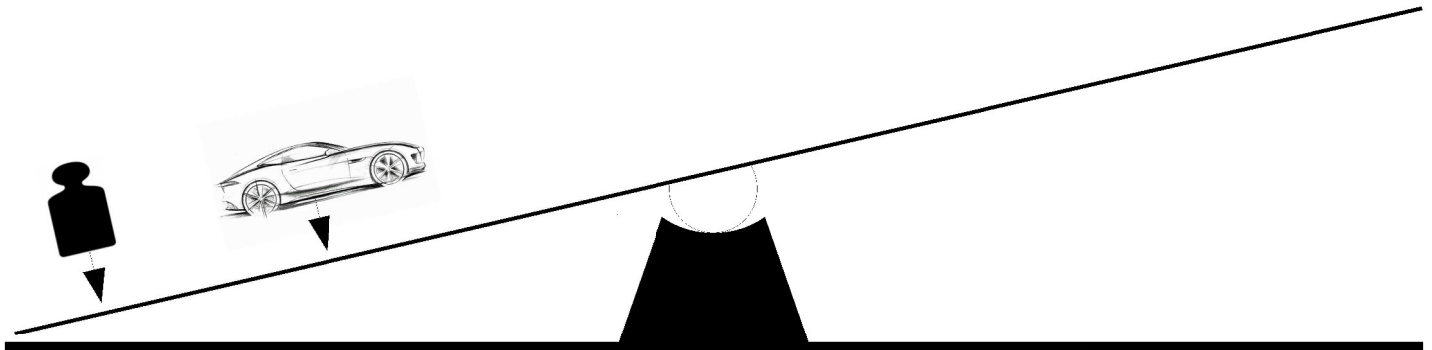


Figure 1: Car and counterweight are placed on the seesaw.

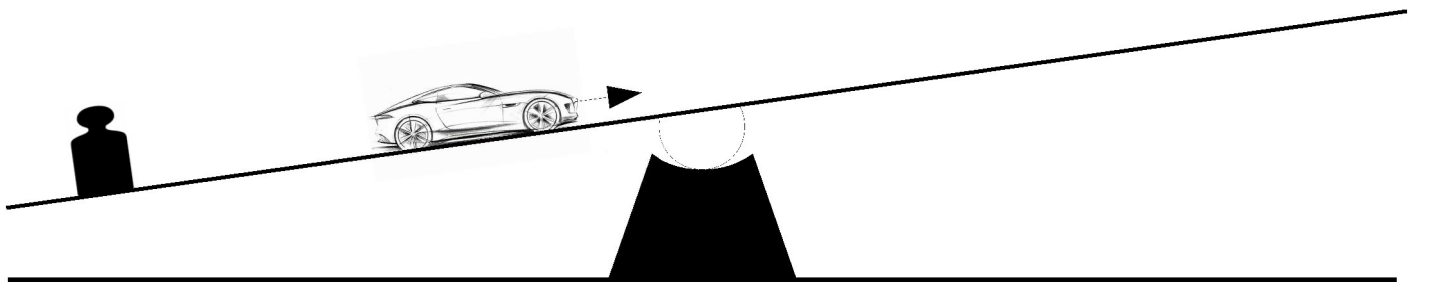


Figure 2: The car starts moving in the required direction.

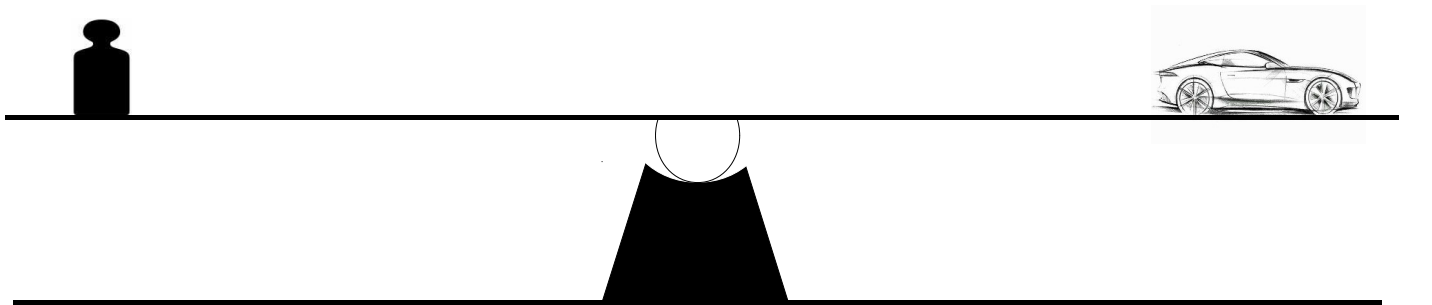


Figure 3: The car stops, when plain alignment is achieved.

b) Specific Test Sequences

To determine the functionality of the system, a few test sequences were specified in an old task the STIEI had for a nation wide contest in 2007.

The first test determines a Line tracking function of the car. It has to be able to drive stable along a black line on the ground/seesaw. Since we are only driving straight on the seesaw the ability to follow sharp curves is not necessary, but the car has to be able to drive forward and backward on the same line.

The second test sequence is done **without** a counterweight. It should show the basic stability of the car on the seesaw:

- 1) The car starts at one end of the seesaw and has to reach the center within 30 seconds.
- 2) Within 60 seconds the car has to align the seesaw for at least 5 seconds.
- 3) The car has to leave the seesaw on the opposite side where it started.
- 4) The entire process has to be completed within 3 minutes.

The third test sequence requires a counterweight. This sequence demonstrates the dynamic capability of the system.

- 1) Same instructions as in point 1 and 2 without the counterweight.
- 2) A counterweight (maximum 20 % of the weight of the car) is placed somewhere on the seesaw. The car has to change automatically its position to align the seesaw in plane direction for at least 5 seconds.
- 3) The entire process has to be completed within 3 minutes.

a) Sensors

Our goal with the sensors is to measure the accurate pitch of the car when exposed to a tilt (in our project; when it is on the seesaw). Since the seesaw only produces a tilt of 11° angle we have to be very accurate especially with small angles. To achieve this goal we decided to use a combination of two sensor types: an Accelerometer and a Gyroscope. This chapter describes the specific sensors we used as well as the filter system and implementation on the micro controller.

Accelerometer

The available Accelerometer is a type MXD2020E/FL from MEMSIC.

Feature list from data sheet:

- Resolution better than 1 milli-g
- Dual axis accelerometer fabricated on a monolithic CMOS IC
- On-chip mixed mode signal processing
- 50,000 g shock survival rating
- 17 Hz bandwidth
- Continuous self-test
- PWM-Output

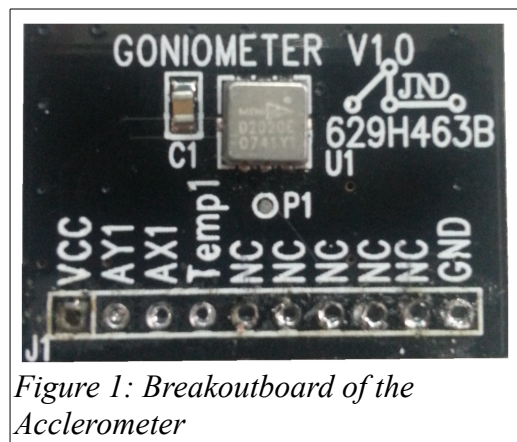
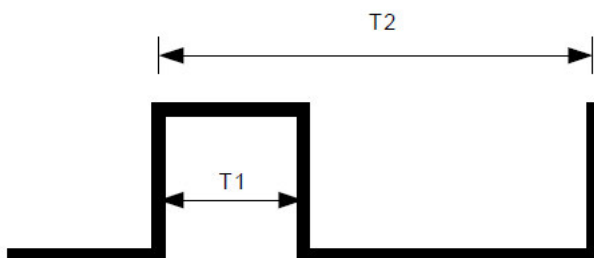


Figure 1: Breakoutboard of the Accelerometer

Raw Data:

The signal the controller receives from the sensor is a 100 Hz Pulse Width Modulated (PWM) signal that represents the acceleration.



$$T2 = \frac{1}{100\text{Hz}} = 10\text{ms}$$

T1 depends on the acceleration the sensor is exposed to.

Figure 2: PWM Sketch

If T1 is measured by the micro controller, the formula to calculate the acceleration in g is:

$$a_{(g)} = \left(\frac{T1}{T2} - 0.5\right) / 0.2 \quad \text{with} \quad \frac{T1}{T2} \quad \text{corresponds to the duty-cycle of the PWM signal.}$$

To get accurate readings from the sensor our micro controller has to measure two times, the High period and the Low period of the PWM signal. The High period time equals T1 and High + Low period time equals T2.

For this purpose we use one Programmable Counter Array module (Module 0) of our controller. Module 0 is configured to detect every transition of the signal (high->low and low->high) from the input on Port 1.0. When this happens, the controller produces an interrupt of the main program and vectors to the PCA_ISR (PCA interrupt service routine). With the help of a Flag (CCF0) it also follows the procedure to extract the High or Low period time.

Module 0 is also configured as a 16 bit counter, so the maximum value it can reach is $2^{16} = 65536$. When this value is reached, the counter overflows and starts counting again from 0 to 65536. The PCA in general operates at a frequency of $\text{SYSCLK}/4$ which is exactly 5.529.600 Hz (~5.5 MHz). Since T2 is always exactly 10 ms we can use this time period to decide if an overflow counter is required:

$$\text{Time for 1 value: } \frac{1}{5529600} \approx 1,808 * 10^{-7}$$

$$\text{Time for 1 overflow: } 1,808 * 10^{-7} * 65536 = 11,85 \text{ ms}$$

One overflow takes more time than the maximum period time of our signal we try to measure. This leads to the conclusion that we don't need to count overflows and can use the counter as direct reference for the times T1 and T2.

Calculations:

To extract an angle from the acceleration data of the sensor a few calculations have to be made.

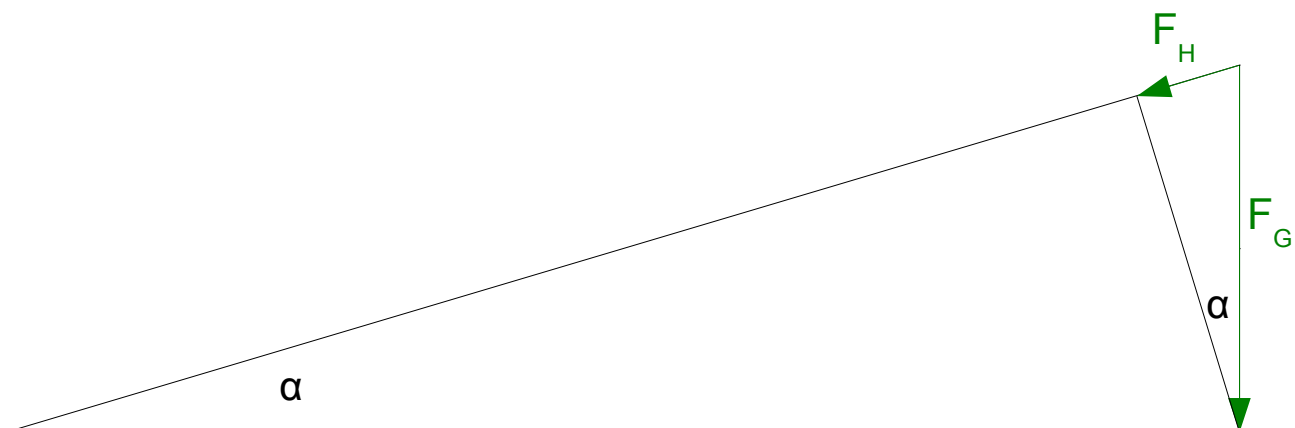


Figure 3: Sketch of the forces applied on the car when standing on the seesaw.

The angle α : $\alpha = \sin\left(\frac{F_H}{F_G}\right) = \sin\left(\frac{m * a}{m * g}\right) = \sin\left(\frac{a}{g}\right)$

Since $g = 9,81 \frac{m}{s^2}$ is a constant, the only variable left is the acceleration a. To simplify calculations for the microcontroller and save processing time we also use an approximation of this formula:

$$\alpha = \frac{a}{g}$$

The approximation leads to accurate measurements between -30° and $+30^\circ$ angle. Everything below/above these angles should be calculated with a sinus function again. In our particular project this is not necessary.

Summary:

Advantage:

- No Drift (Stationary accurate)

Disadvantage:

- Every acceleration of the car is detected as “angle” (=> unstable)

Gyroscope

We decided to use a L3GD20 from STMicroelectronics.

Features from the data sheet:

- Three Full Scale axes (roll-pitch-yaw)
- Selectable resolutions ranging from 250 to 2000 degrees per second
- I²C or SPI interface with 16-bit output resolution
- Integrated temperature sensor
- Integrated high- and low-pass filters with selectable bandwidth

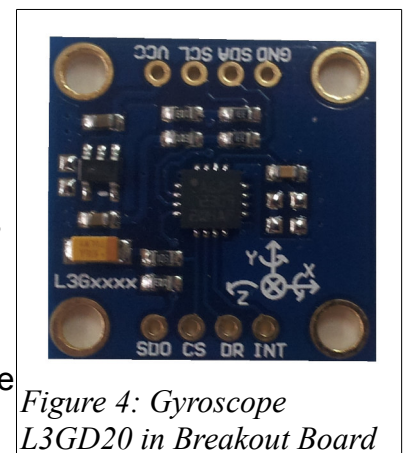


Figure 4: Gyroscope
L3GD20 in Breakout Board

A Gyroscope gives information of the orientation with the help of angular velocity. This means the sensor measures the actual speed it is rotating around an axis. This angular velocity can be multiplied by the time, so it is possible to calculate the angle it changed.

Communication:

To extract the Raw Data from the sensor we used the communication interface SPI Bus. It uses 5 Pins on the microcontroller:

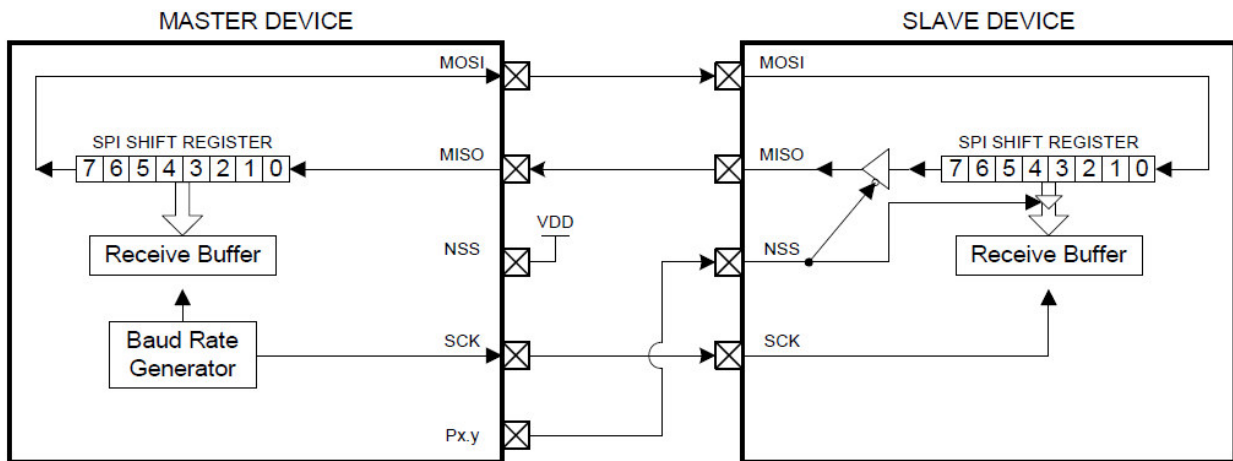


Figure 5: Sketch of SPI Bus Communication with 1 Slave and Full Duplex Operation

1. CS (NSS in Figure 5) → Every Slave has its own Chip Select line connection to the master when using a SPI Bus system. When a Slave is selected (active high or active low, depends on the slave) Every transmission over the MOSI or MISO lines will be transferred between this particular Slave and the Master.
2. MOSI → The Master Out Slave In line is connected to the Master and every Slave and transmits data from the Master to the selected (CS) Slave.
3. MISO → The Master In Slave Out line is the exact opposite of the MOSI line and transmits data from the selected Slave to the Master.
4. SCL (SCK in Figure 5) → The System Clock Line supplies all Slaves with the frequency of the Master.
5. DR → The Data Ready line is an additional connection to the microcontroller that indicates when there is new data available on the sensor.

While the connections 1-4 are the standard connections used for an SPI Bus System, the fifth line is a special feature of the L3GD20. It produces an interrupt in the main program like the accelerometer when new gyro data is available and vectors to the PCA_ISR. With the help of a different flag (CCF3) the program follows the instructions to calculate the latest change of the angle the sensor (and with it the car) is exposed to. While this system uses a lot of Pins on the controller compared to other communication protocols, the advantage lays in a very fast procedure, not much processing time and an easier implementation on the controller.

Configuration:

When the SPI Bus communication works, the correct configuration of the sensor is very important. The L3GD20 has a lot of registers that can be set up before using the sensor. The most important ones are the 5 control registers, the 2 FIFO control registers and the output registers. For more information on the registers and their purpose please see the data sheet of the L3GD20.

Here is a quick overview of the configuration we use:

- **Sensitivity: 250 °/s (or dps)**

Control Register 4: FS1-FS0

In general the following rule applies: If the Sensitivity is lower, the sensor gets more precise.

With 250 dps we measure with an accuracy of $\frac{250 \text{ dps}}{2^{16}} = 3,18 * 10^{-3} \text{ dps}$.

This means for example when our output value from the Gyroscope changes from 400 to 401 in one period of time the angle changed for $\sim 3 \text{ m}^\circ$ in that time.

- **Output Data Rate: 95 Hz**

Control Register 1: DR

The speed of the output always depends on the project. When there are fast changes of angle (for example with a quadrocopter) the output data rate should be fast enough for it. Since we only have changes of a maximum of 10° (from -5° to $+5^\circ$) in more than one second, we can choose the lowest possible rate. This also lowers the noise the sensor produces. And since the integration of our angular velocity leads to the actual angle, noise is also added to this calculation. This effect is called drift and leads to more and more inaccurate measurements over time. With slow output data rates we can lower this effect, but it can only be eliminated completely when the Gyroscope is used in combination with our Accelerometer.

- **Data Ready Interrupt**

Control Register 3: I2_DRDY

If this Interrupt is enabled, the sensor produces a high signal on its DR Pin, when there is new data in the Output Registers available. Reading the all (!) Output Registers resets the Interrupt and it occur again with the next data arrival. Note: This Interrupt is produced very precisely with 95 Hz, which is the Output Data Rate.

There are a couple more options to set up the sensor. The first options allow to store up to 32 bytes of measured data in the sensor unit and then read it all at once, when the memory (FIFO) is full. We decided to read every time when new data is available, because the SPI Bus is very fast and the output rate of the sensor is low.

The second number of options contain a more complex threshold interrupt system of the sensor. It allows to produce interrupts, when the speed of the sensor reaches certain levels. This is not necessary in our project and details can be seen in the data sheet.

Calculations:

The output registers mentioned earlier are transmitted as 16 Bit values for each axis measured. They represent the angular velocity in a certain time period. The angular velocity can then be calculated with this formula:

$$angular\ velocity = \frac{Sensitivity}{2^{Resolution}} * value = \frac{250\ dps}{65536} * value$$

Integration of angular velocity leads to an angle in degrees.

$$angle_{(degree)} = \int angular\ velocity * dt$$

Summary:

Advantages:

- independent from accelerations
- very fast

Disadvantage:

- Every integral sums up the noise error of the measurement → unstable and unreliable (Drift)

Result:

One of the first things we had to do in our project was to set up each sensor individually. A comparison of the two sensor types summaries and practical tests with each sensor standalone leads to the conclusion, that a combination of this sensors could eliminate theoretically the disadvantages of the sensors, while advantages remain.

We decided to use a relatively simple method compared to others, called a "Complementary Filter".

b) Complementary Filter

We get two signals which theoretically should be the same. The reasons, why they aren't, are told in the sensors description. So it is clear, none of the sensors is able to measure the angle correctly standalone. That's why we are using a complementary filter, to combine the benefits of both sensors and eliminate their disadvantages.

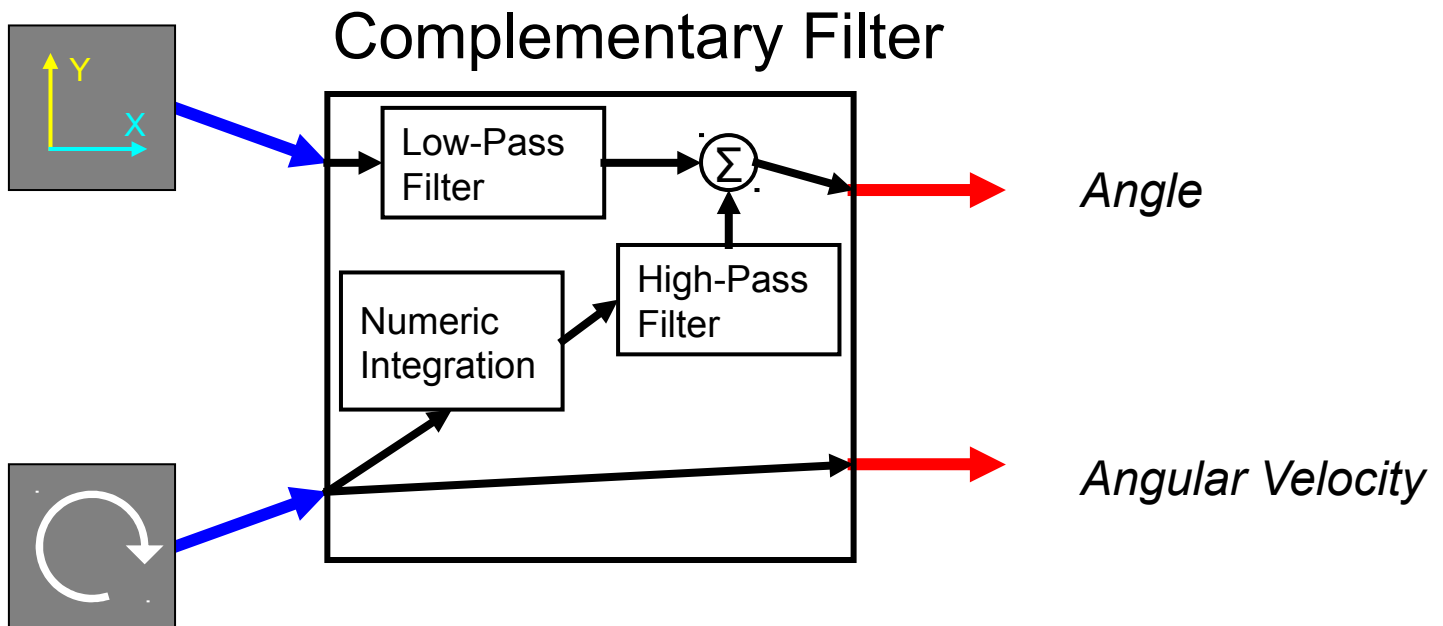


Figure 6: Basic Principle of the Filter

Accelerometer with Low-Pass Filter:

In general a low pass filter filters out the fast, high frequent changes in a signal. In our case it filters the acceleration provided by the accelerometer. The fast changes of this value are the result of real acceleration of the cars movement and vibrations the sensor is exposed to. With the low pass filter we lower the effect of the main disadvantage of the sensor, but we cannot eliminate it entirely.

Gyroscope with Numeric Integration:

The readings of our gyroscope are degrees per second. As described in chapter 2 a) we can use integration of this value to convert it to an angle. The time between two readings is always fixed to 10,52 ms (1/95Hz), this makes the integration in the program even easier. Every value measured by the program is simply added to the angle itself. This leads to an accurate measured angle with the typical drift mentioned earlier.

Gyroscope with High-Pass Filter:

A high pass filter is the exact opposite of the low pass filter. It blocks long term, or low frequent changes of the measured value. The low frequent changes of the Gyroscope is the drift it produces after the integration of the value. After the high pass filter, almost only the fast accurate changes of Gyroscope while rotating remain. Like with the accelerometer, the main disadvantage of the Gyroscope can only be lowered with this technique, not eliminated.

After this filtering the two signal have to be combined to lead to the desired angle.

Implementation as code in C:

Luckily the implementation of this filter in the program is fairly easy compared to other filter methods like the Kalman Filter system. It is done with one single line within the code:

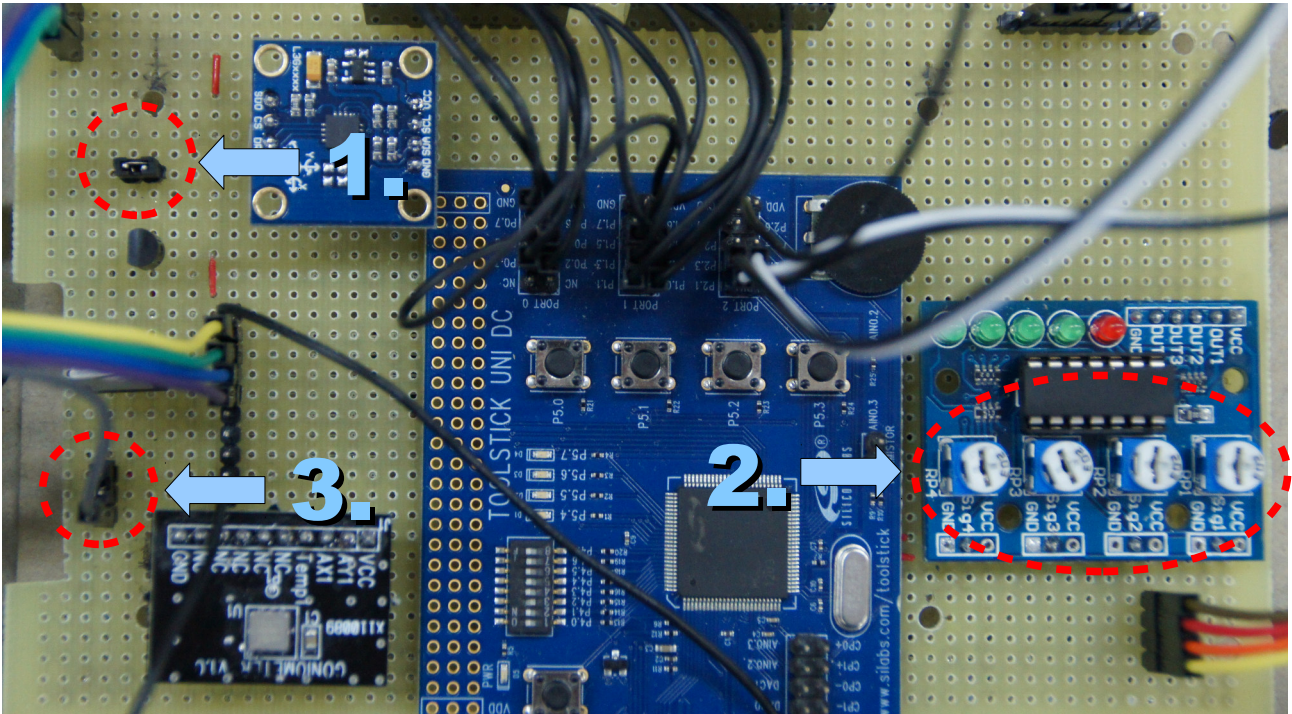
```
angle = 0.98 * (value_gyroscope + angle) + 0.02 * value_accelerometer;
```

This line is executed within the program loop every time when the Gyroscope sends new data (DR Interrupt).

- **“0.98 * (...)” High-pass-filter**
98% of the Gyroscope value is added every time this line is executed in the program loop. When the car remains stationary (= no rotation) this value becomes very low and does not change the angle value very much. The only changes that remain is the drift.
- **“0.02 * value_accelerometer” Low-pass-filter**
This part of the formula adds only 2% of the accelerometer value. Only over a longer period of time the accelerometer value wins over the Gyroscope value. So when the car is stationary the measurement of the accelerometer corrects the drift of the Gyroscope.
- **“value_gyroscope + angle”**
This is the numerical integration of the Gyroscope raw data in degrees per second to degrees.

a) Manual

Setting up the car:



Before the car can be started properly it has to be set up with these steps:

1. Set the jumper for 3.3 V power supply.

This jumper connects the controller and the sensors to 3.3 V from the voltage regulator. It activates the Line Tracking Diodes and pulls the PWM contacts of the motor driver to low level. Without this the motors will drive uncontrollably until the controller is plugged in, because they are floating (either logic low or logic high).

2. Adjust the intensity of the Infrared Diodes for Line Tracking.

When the car stands on the ground, all green LEDs on the Line Tracking sensor module should light up. When the car stands on the ground and the diodes are above a black surface the corresponding LED should stop lighting up.

3. Plug in the 5 V power supply for the motor driver.

At the end the motor driver should be supplied from the 5 V voltage regulator to enable the motors.

After this quick setup the car is ready to go and can be controlled with the buttons and switches on the controller board. This is described in the following pages.

The car has 3 different operation modes:

1. *Balancing sequence:*

- The car has to be aligned towards the seesaw (up to 45°).
- It starts driving towards the seesaw until a black line is detected.
- The line tracking mode lets it drive upwards the seesaw for ~5 seconds. This time should be enough for the car to be near the middle of the seesaw.
- It then starts balancing until plain alignment is achieved.
- It waits for 5 seconds.
- It drives off the seesaw (line tracking mode enabled).
- When the front sensors don't detect a line anymore the line tracking is disabled and the car drives for another 5 seconds away from the seesaw.

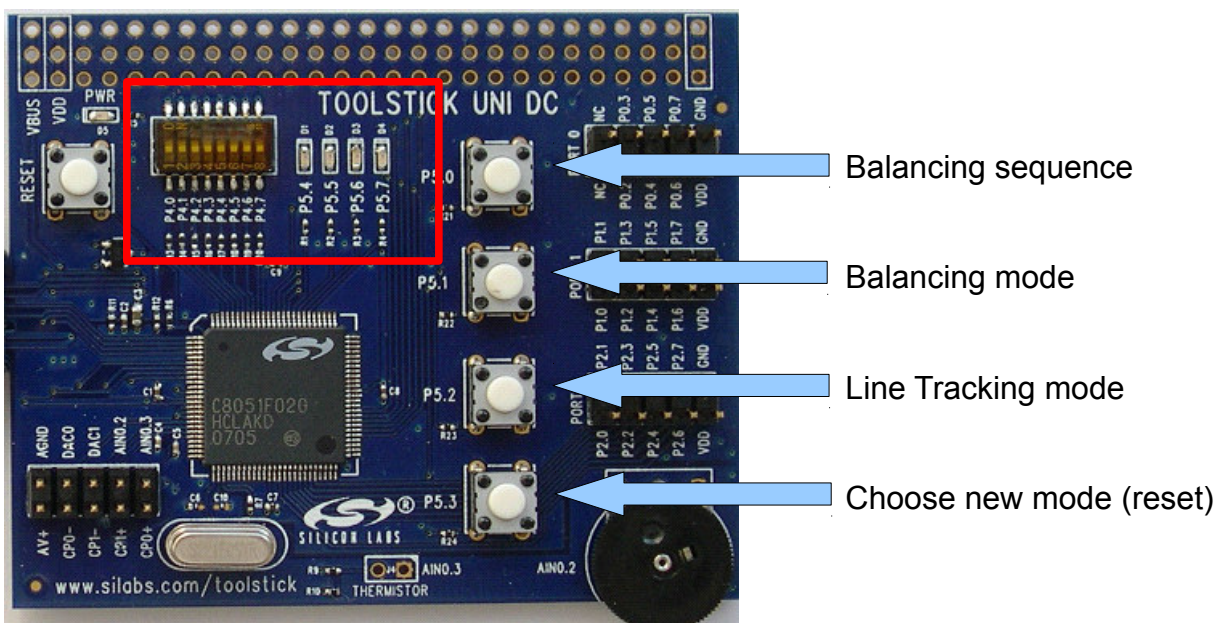
2. *Balancing mode:*

The car executes only the balancing mode and remains in the correct position to align the seesaw. Every change to the seesaw (counterweights) result in movement of the car to a new position.

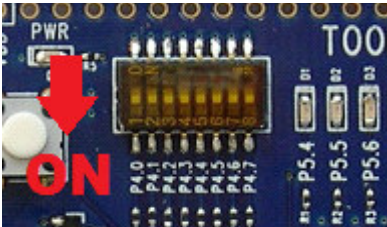
3. *Line Tracking mode:*

The car follows a black line on the ground and follows curves up to 80° of angle.

These 3 modes can be selected with these buttons on the controller itself:



It is possible to choose the different control modes (P, PD, PID...) of the control algorithm via the control switches P4. Only one switch should be set ON at the same time. More switches in ON position normally lead to the last active switch to decide which mode is selected.



- P4.0: P
- P4.1: PI
- P4.2: PID
- P4.3: PD

P and PI mode should result in the car oscillating around the balance point and it is not able to reach and stay in the correct position. This is recommended for demonstrating the importance of the Derivative part of the algorithm in this particular project.

PID mode is most accurate mode and is recommended for showcasing the car.

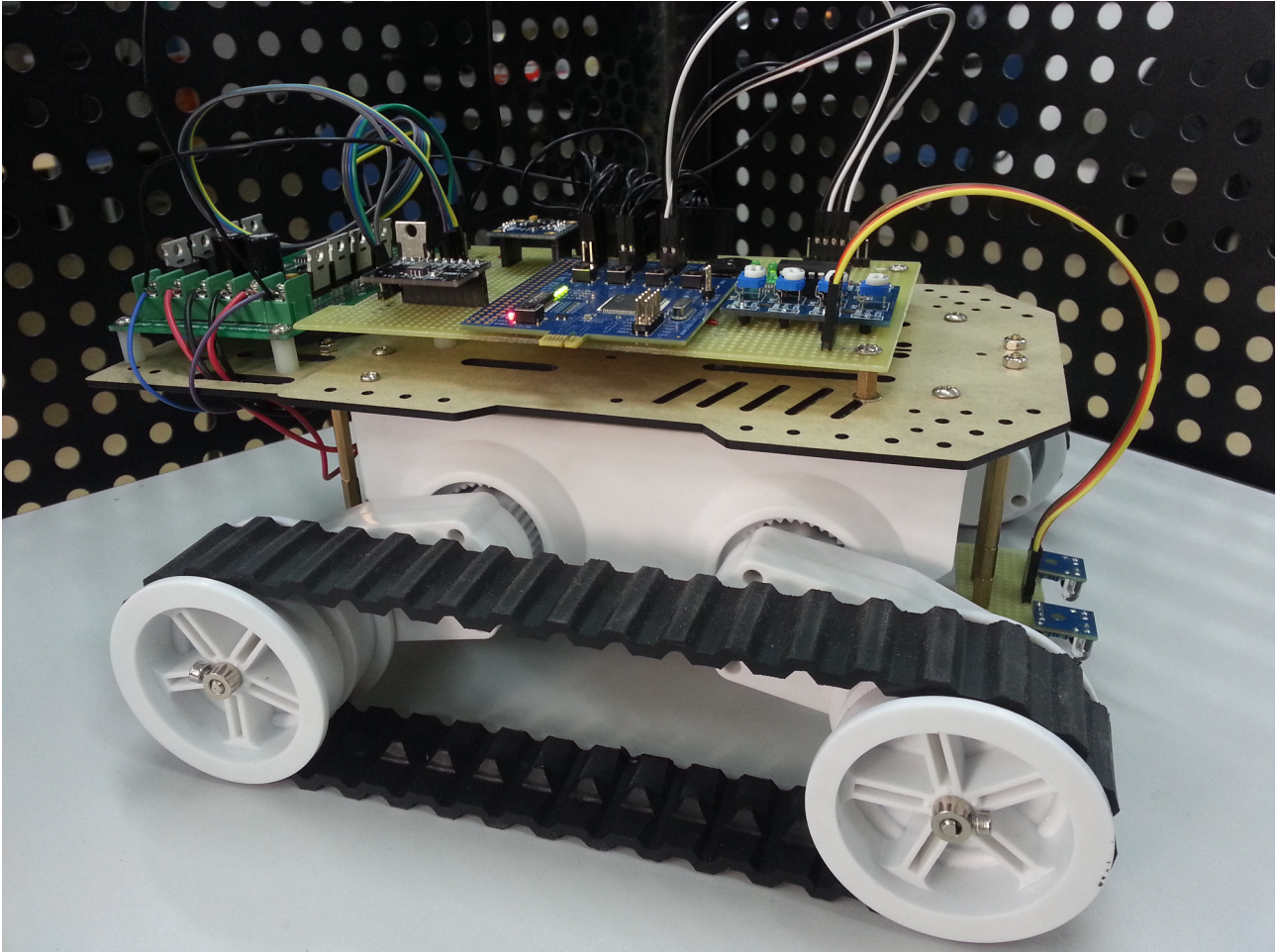
PD mode is most of the time the fastest mode, but can lead to inaccuracy. This could go so far that the balancing sequence cannot be completed and the car stands still on the seesaw.

Additional Feature:



The LEDs located on the controller board light up when the angle measurement detects an angle between -1° and $+1^\circ$. It also indicates the 5 seconds waiting time when the car is aligned on the seesaw. While the Balance sequence is running the LEDs are a good indicator to detect why the car is not leaving the seesaw.

b) Chassis



The car

- Dagu Rover 5
- 2 independent DC motors
- Optical encoders
- 1:86 gear ratio

The motor driver

- 2 H-Bridges
- PWM-Inputs
- Direction Pins
- 10A continuous current

c) Motor Driver

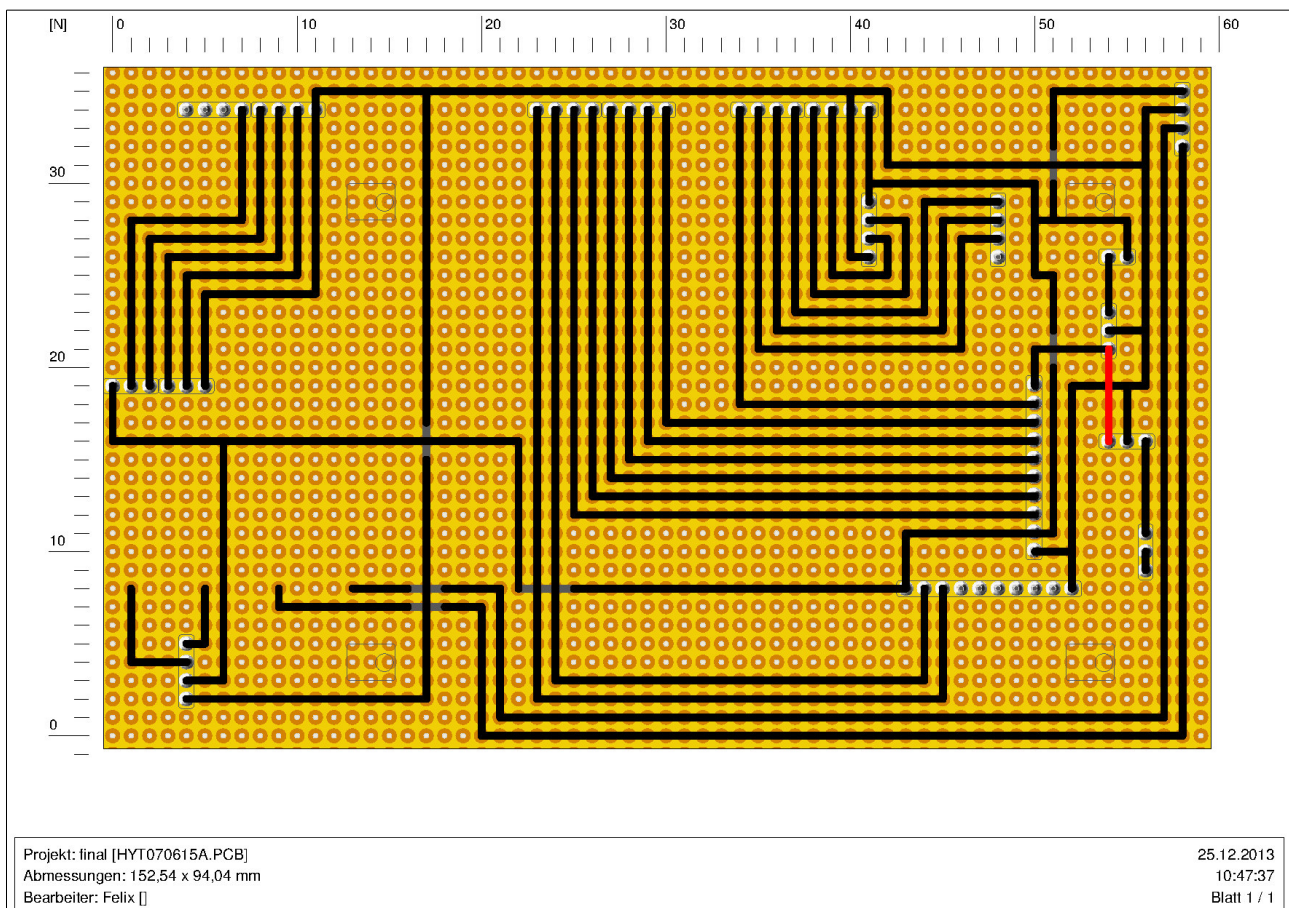
Explanation:

The motor drivers logical unit has 6 connections, two of them are for the +5V power supply. Then it has the pins PWM1, DIR1, PWM2 and DIR2.

The direction pin inverts the direction in which the current flows. The PWM is, if not connected, in high state which means it is running all the time. If a PWM signal is connected, the pin gets pulled down during the low-time of the PWM signal. In this phase, the motors power transistors are blocked. With this procedure, the PWM control can be applied to the motor driver.

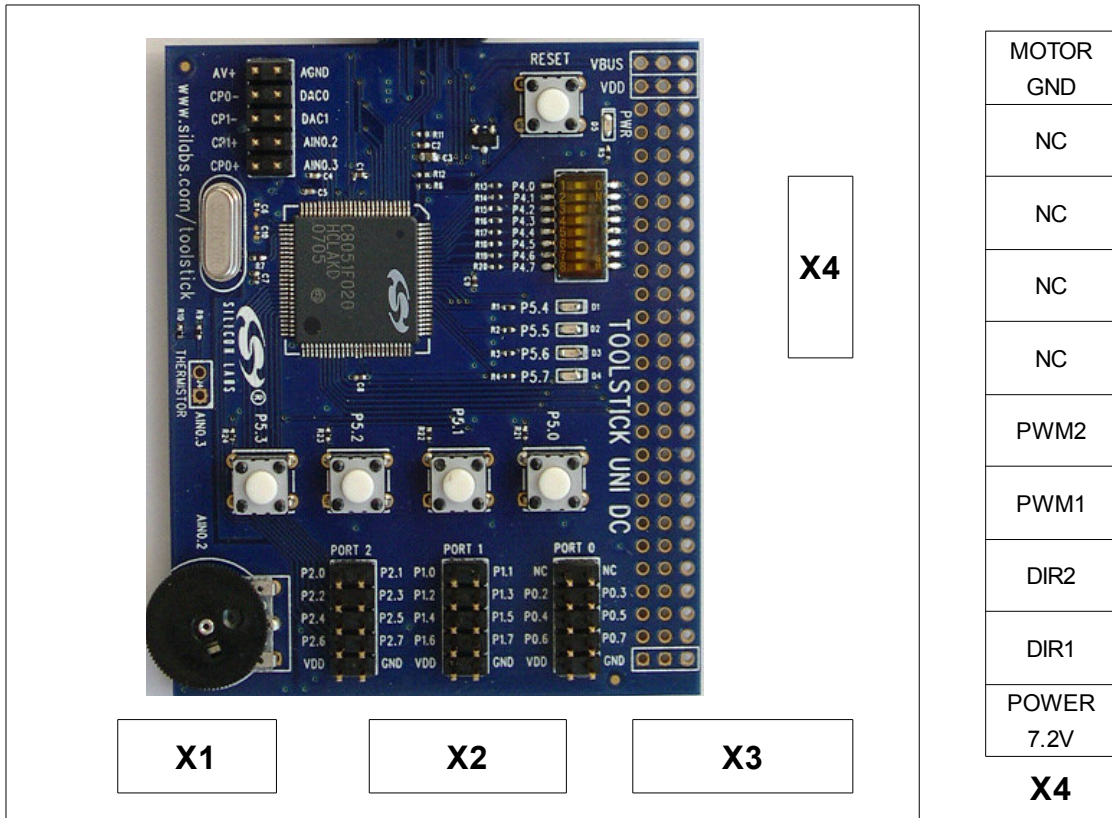
The motor drivers power unit has also 6 connections, two for the power supply the motor needs. In this case, we use a 6x 1.2V battery pack, so there are is a voltage of 7.2V. In addition, there are 2x MOTOR1 and 2x MOTOR2 pins, where the motors are connected.

The circuit board

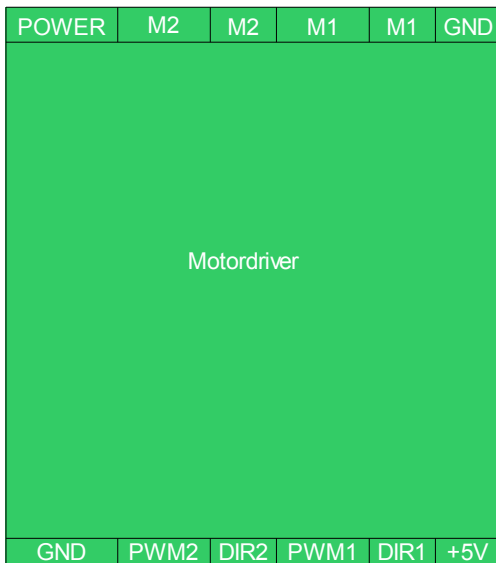


The circuit board connects all of the sensors and the peripherals with the microcontroller. It was first designed with the “Lochmaster 4” software and then soldered. Several changes were necessary until its final form.

d) Pin Assignment



X1	NC	NC	NC	P2.1	P2.0	P2.2	P2.3	NC
X2	P1.0	NC	NC	NC	NC	P1.2	P1.1	P1.6
X3	P1.5	P1.3	P1.4	P0.3	P0.2	P0.4	P0.GND	P0.VDD



a) Sensors

To hold the car from falling off the seesaw, we decided to use line tracking sensors. It is possible that the car has to drive backwards so we used 2 sensors at the front and 2 at the rear of the car. Each sensor unit has an Infrared Light Emitting Diode and one light sensing Diode.

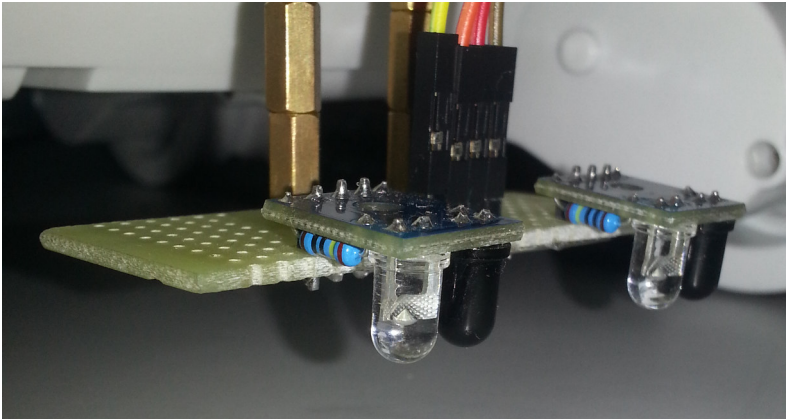


Figure 1: Sensors at the Front of the car

The Infrared LED (Figure 1) emits infrared, non visible light. The black Diode (Figure 1) detects light in the particular spectrum the Infrared LED emits. This should prevent the sensor from detecting daylight and only react to the light emitted by the sensor itself. Of course this does not work perfectly and with brighter daylight the sensors tend to false readings. To avoid this we put the sensors as near as possible to the ground.

The intensity of the Infrared LED can also be altered with the electronics shipped with the sensor array:

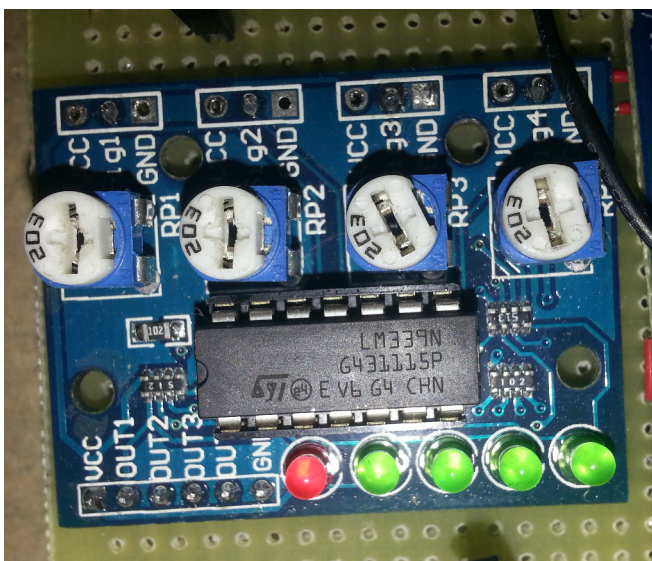


Figure 2: Sensor Electronics

Every element of the sensor array can be configuration individually with the potentiometers shown in Figure 2.

It is possible to adjust them to a new lighting situation.

These electronics also give precise output signals to the micro controller:

0 V	3.3 V
Reflection	No Reflection

b) Procedure

In a normal program sequence every chain gets the same percentage of power (duty-cycle of PWM) either from the control algorithm or from a fixed value. This percentage is given to an output program that directly generates the PWM signal for the motor driver. The only way to steer the car is to alter one individual chain power separated from the other one. This is achieved by adding an additional program between these two programs:

```
// PD Algorithm
pwm_l = pwm_r = control_algorithm();

// Steering
pwm_l += steer(pwm_l, left, 10);
pwm_r += steer(pwm_r, right, 10);

// Output
Controller_Output(pwm_l, pwm_r);
```

By constantly calling this sequence the steering can effect the output always as needed.

A closer look at `steer()`:

- *Declaration:* `char steer(char PWM, bit side, char offset);`
- *Input values:*
 1. **PWM** – The desired acceleration in %
 2. **side** – This bit determines the chain that has to be checked for steering
 3. **offset** – This value adjusts the impact of the steering. A higher value leads to stronger steering until the car rotates on its position (`offset = -PWM`)
- The Return value is 0 when the opposite sensor detects no reflection and (+/-)**offset** if the opposite sensor detects a reflection. The sign depends on the direction the car is moving (positive or negative PWM).

Examples:

```
pwm_l += steer(70, left, 10);
```

If the sensor on the right front (PWM is positive and left chain is selected) detects a reflection, the return value is -60 and the PWM of the left chain is lowered to 10.

```
pwm_l += steer(70, left, -70);
```

If the sensor on the right front (PWM is positive and left chain is selected) detects a reflection, the return value is -140 and the PWM of the left chain is lowered to -70.

About the Manufacturer

The C8051 F020 is manufactured by the US company Freescale.
Freescale arose from Motorolas semiconductor department, which was spun off in 2004.

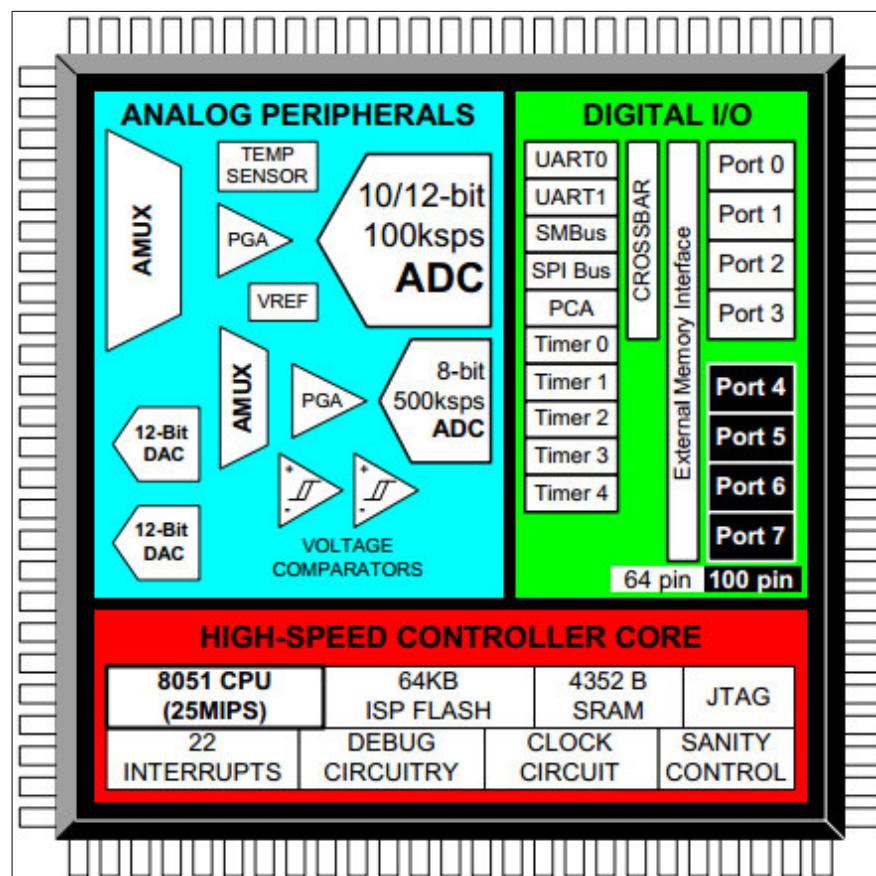
The company focuses on the automotive, consumer, industrial and networking markets with its product portfolio including microprocessors, microcontrollers, digital signal processors, digital signal controllers, sensors, RF power ICs and power management ICs. In addition, the company offers software and development tools to support product development. ⁽¹⁾

About the MUC

The used MUC belongs to Freescales 8-Bit family and uses the popular 8051 architecture, it offers several on-chip benefits such as different timers, ADCs/DACs and bus-interfaces (I²C, SPI).

The C8051 F020 is a good solution for the given task, as the controller is able to connect to many different sensor types like analog sensors, digital sensors, sensors with PWM outputs or bus-connected sensors via I²C or SPI.

Block diagram from datasheet:



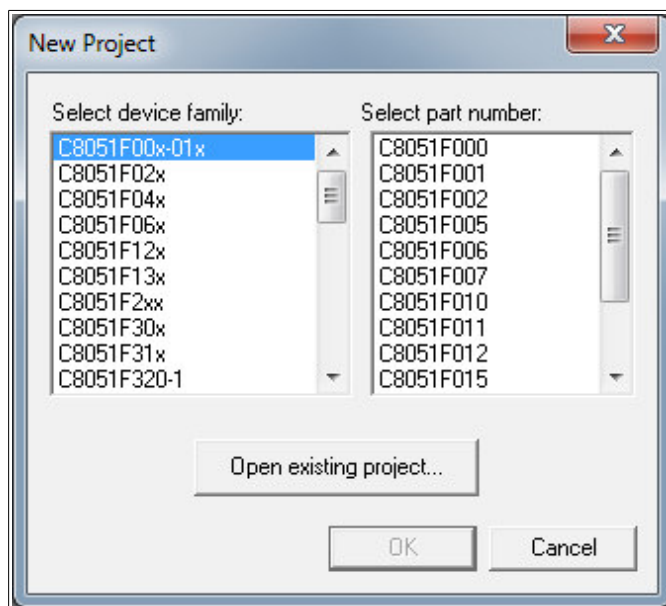
(1) http://en.wikipedia.org/wiki/Freescale_Semiconductor 12.09.2013 10:12

About Configuration Wizard 2

The Config2-Tool is one of the reasons, why it's never a bad idea to choose a SiLabs controller. The Configuration Wizard 2 is a software with a GUI, that allows to easily generate the code for the controller initialization, depending on which controller you are using exactly.

On the following pages, the functionality of the tool will be demonstrated.

Step 1: New Project



1. Select device family: C8051F02x
2. Select part number: C8051F020

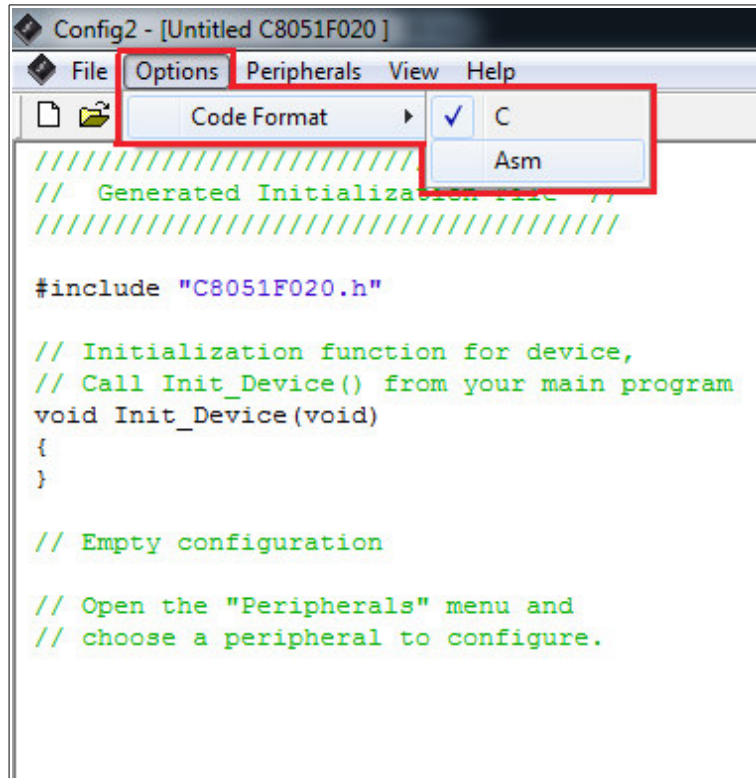
Save Project / Code

Save Project: File → Save Project As...

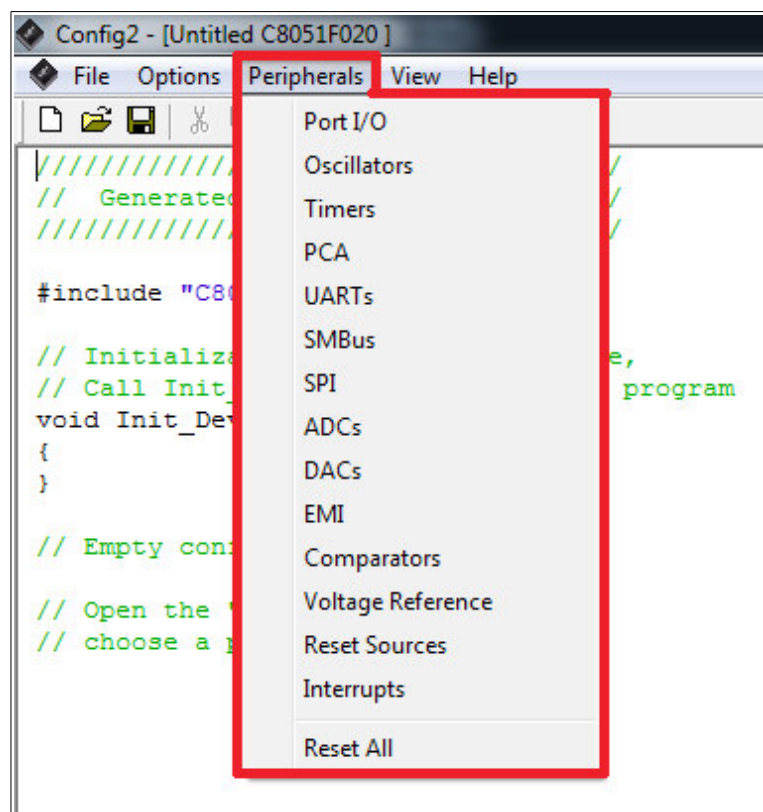
Save Code: File → Save Source As...

Step 2: Select Code Format

It is possible to generate the code in C or Assembler.



Step 3: Configure Peripherals



The Crossbar

Priority Crossbar | Ports 4/5/6/7 Output

Crossbar Priority Decoder Setup

Enable Crossbar
 Disable Weak Pull-Up

■ Assigned
■ Skipped
 D - Digital
 A - Analog
 O - Open Drain
 P - Push-Pull

Pin I/O	P0								P1								P2								P3							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
UART0																																
TX0																																
RXD																																
SCK																																
MISO																																
MOSI																																
NSS																																
SDA																																
SCL																																
TX1																																
RX1																																
CEX0																																
CEX1																																
CEX2																																
CEX3																																
CEX4																																
ECI																																
CP0																																
CPT																																
Timer 0																																
/INT0																																
T1																																
/INT1																																
T2																																
T2EX																																
T4																																
T4EX																																
/SYSCLK																																
CNVSTR																																

Analog / Digital →
 Push-Pull / Open Drain →
 ALE / RD / WR
 Enable Low Port EMI

n = Pin Number within the Port Selection
 AIN1.n/Ain+8
 ADn/Dn

PTMDIN = 0x0F;
 PTMDOUT = 0x04;
 PTMDOUT = 0xC0;
 XBR0 = 0x83;
 XBR1 = 0x2A;

Priority Crossbar | Ports 4/5/6/7 Output

Ports 4/5/6/7 Push-Pull/Open Drain Selection

	P4								P5							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Push-Pull / Open Drain -->	0				0				0				0			
	WR RD ALE															

	P6								P7							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Push-Pull / Open Drain -->	0				0				0				0			

O - Open Drain
P - Push-Pull

```
P1MDIN = 0x0F;  
P0MDOUT = 0x04;  
P1MDOUT = 0xC0;  
XBR0 = 0x83;
```

Explanation:

This are the selected peripherals. Check them in the box.

Here you can choose, if you need Analog/Digital Ports and if they have to be Push-Pull or Open-Drain. Click on the letter in the little box to toggle the modes.

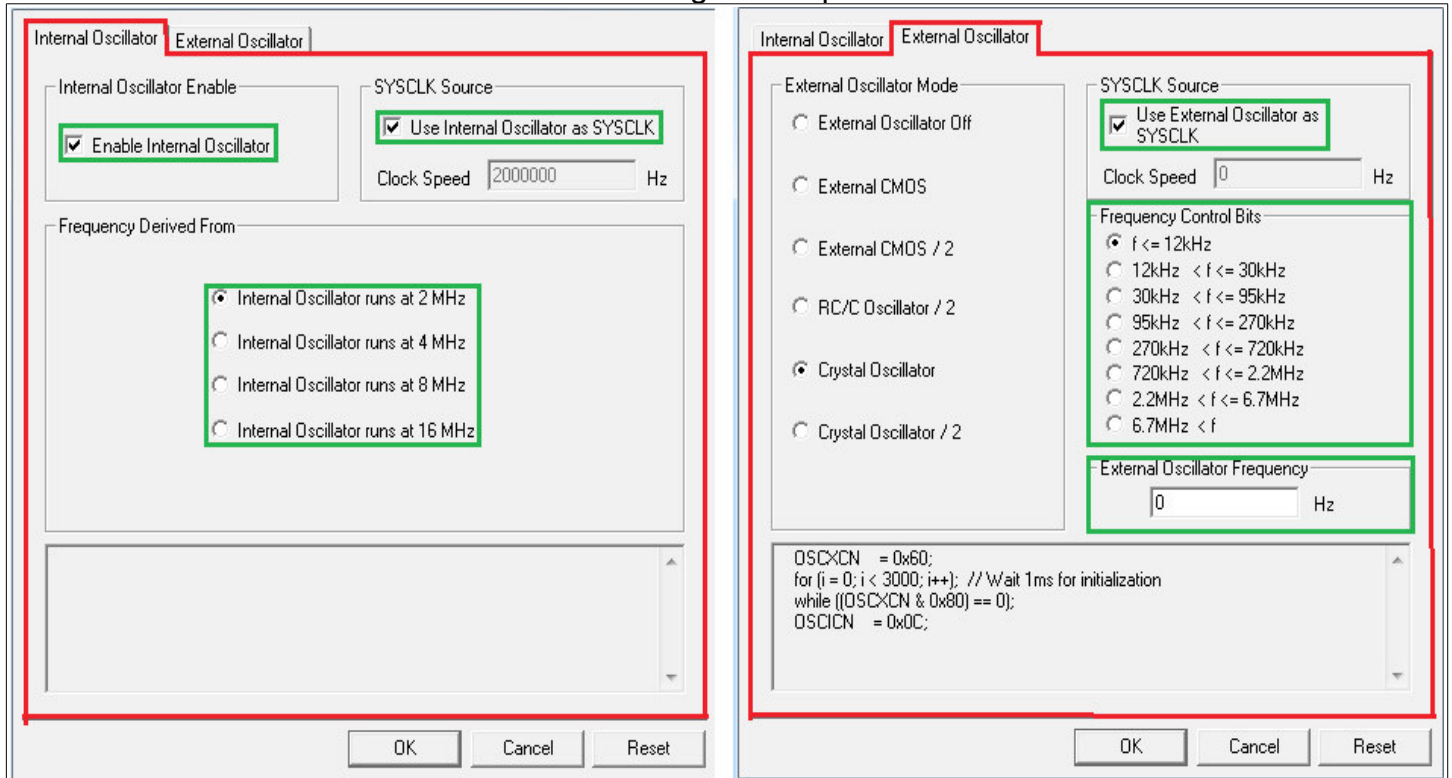
This is the pin of the port to which the peripheral is connected. If you toggle a pin as analog, it will be reserved and not used for peripherals as seen by P1:4-7.

Here you can enable the crossbar (always necessary!) and disable the weak Pull-Up resistors.

This is the generated code.

Oscillator

It is possible to operate internal or external oscillators. In the following figure you can see, which settings are important:



Programmable-counter-array (PCA)

A very important component of our project is the PCA. The PCA consists of a 16-bit timer and five capture-compare modules. It can be used for many different tasks.

For example:

- Capture on rising edge / falling edge / transitions
- Software Timer
- Highspeed Output
- Frequency Output
- 8-bit or 16-bit PWM

In our project, we used the capture on rising edge for the Data-Ready interrupt, the capture on transition for the PWM-input-measurement and the 8-bit PWM outputs for the motor control.

Configuration:

- PCA0: Enabled, SYSCLK/4
- Module 0: Capture on transition
- Module 1: 8-Bit PWM
- Module 2: 8-Bit PWM
- Module 3: Capture on rising edge
- Module 4: None

Interrupts

The screenshot shows the 'Interrupts' configuration window with the following settings:

- Enable Interrupts:** Global Interrupt Enable (unchecked)
- UART Interrupts:** Enable UART0 Interrupt (unchecked), Enable UART1 Interrupt (checked)
- Timer Interrupts:** Enable Timer 0 Interrupt (unchecked), Enable Timer 1 Interrupt (unchecked), Enable Timer 2 Interrupt (unchecked), Enable Timer 3 Interrupt (unchecked), Enable Timer 4 Interrupt (unchecked), Enable /INT0 Interrupt (unchecked), Enable /INT1 Interrupt (unchecked)
- PCA Interrupts:** Enable PCA0 Interrupt (checked)
- ADC Window Comparison Interrupts:** Enable AD0WINT Interrupt (unchecked)
- External Interrupts:** Enable EX6 Interrupt (unchecked), Enable EX7 Interrupt (unchecked), Enable XTLVLD Interrupt (unchecked)

At the bottom, the register values are displayed:

```
IE = 0x01;  
EIE1 = 0x08;  
EIE2 = 0x40;
```

Buttons: OK, Cancel, Reset

Global Interrupt Enable

No interrupt is working, if this flag is cleared, but the box is not checked, because it will be set after the controller initialization with the software (EA=1;)

UART Interrupts

The UART1 Interrupt is enabled, so it is possible to read incoming data from the serial interface.

PCA Interrupts

The PCA Interrupt has different tasks, it is used for PWM-Input-measurement and it signals, if there is new gyro data.

Control Algorithm

The control algorithm that places the car in the right position on the seesaw can be altered between a PD or PID algorithm. Both algorithms have their advantages and disadvantages. To get a better understanding of this we will describe every element of the algorithm for its own.

The main goal of the algorithm is to set an appropriate output value for the motor. The formula for this is very simple:

$$\text{output} = K_p * p + K_i * i + K_d * d$$

p stands for proportional and contains the difference between a certain set point (normally 0) and the actual angle of the car. ($p = \text{set point} - \text{angle}$)

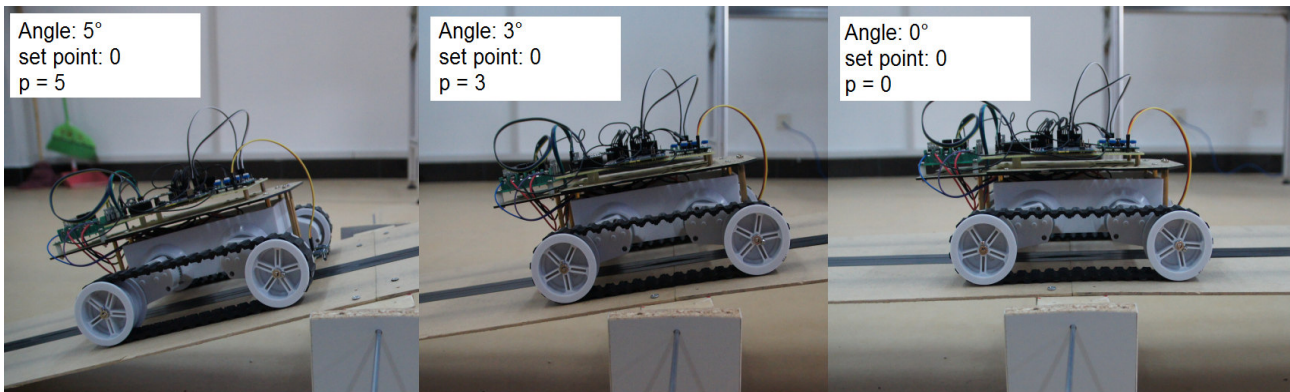


Figure 1: p value for different angles

i stands for integral and contains the summation of differences between a set point and the actual angle of the car. If the car stands for 3 periods in a 5° angle (with set point = 0), i will be 15. The i part grows bigger and bigger over time if an error stays in the system.

d stands for derivative and is higher than 0 if the seesaw is rotating. The d part also counteracts the actual movement from the other part of the algorithm. The d part is the “break” of the system and without it the car will never be able to stop at the right time on the seesaw.

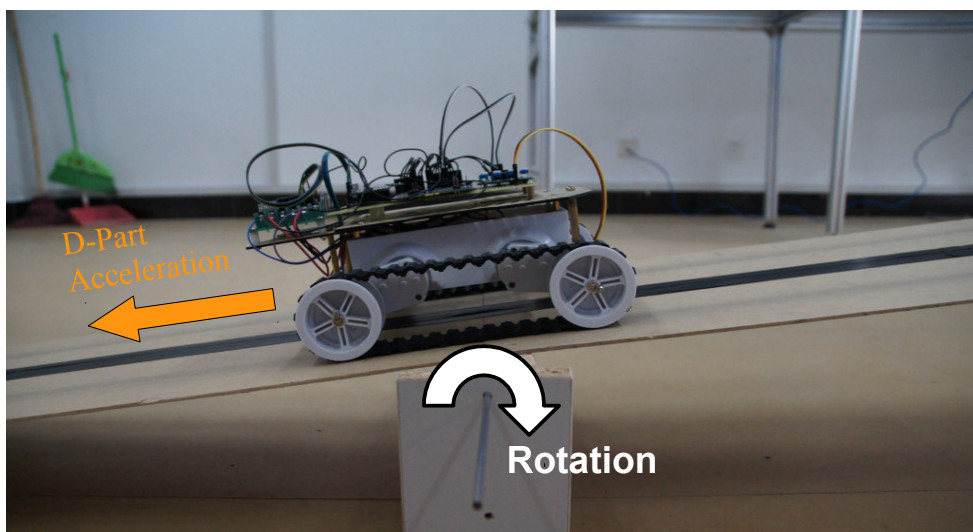


Figure 2: Functionality of the Derivative part

Parameters:

The parameters improve one element over the other and are chosen by testing. Every system has different parameters since it behaves differently. In this part we want to explain the procedure we were going through to find out the best parameters.

1. *Proportional (K_p):*

Since our car can drive up a 5° slope with less than 20% power we find out a good value for K_p would be ~ 5 at first. A 5° slope leads to an output value of 25. With the Integral part later in the project we decided to lower the value to 4, because they both accelerate in the same direction.

2. *Derivative (K_d):*

The derivative part is normally the derivation of the angle: angle d/dt . The derivation of an angle is the angular velocity (angle / time). Since the angular velocity is directly available from our gyroscope we decided to use this value instead of calculating it (which saves processing time). The direct value is very low, because our rotation velocity in this project is very low. To compensate this, K_d has to be very high (>900). The value 1000 lead to the best results in practical tests.

3. *Integral (K_i):*

We decided to add an integral part because we wanted to add more accuracy to our project. The integral part sums up the difference of the angle to the set point. The longer the car is in an incorrect position the higher the integral part gets. But since the integral part makes the whole process slower it has to be limited ("anti-windup"). This limit is set to 20 and K_i is set to 0.4 which leads to a maximum of $0.4 * 20 = 8\%$ power the integral part can deliver. If the p part is nearly 0 (= correct position) 8% power from the integral part are not enough to start the car again and it stands still on the correct position.

a) Functionality of the car

The following results could be achieved in this project. Everything was tested and also documented via video:

[Balancing](#) (YouKu.com)

[Controls Comparison](#) (YouKu.com)

Line Tracking:

The car follows a black line on light ground (~3-4 cm thickness) when balancing on the seesaw. The car also has a Line Tracking mode where its only purpose is driving with ~70% power. In this mode it follows curves up to an angle of ~80° because the chains steer counter-wise like a tank when in Line Tracking mode.

The downside of the Line Tracking are the sensors. Daylight or too much light in the room in general can interfere with the sensors and can cause false movements of the car. But with slightly dimmed light the car worked fine.

PWM Output:

The motors had very good gears and could easily drive up inclinations of ~50° while the control algorithm was active. It was only limited by the grip of the chains and it would be possible to improve it by using another seesaw material for example.

Control algorithm:

The final set of parameters lead to a control time of ~5 sec to balance the seesaw. The whole sequence took about 18 sec and contained:

1. Driving up the seesaw
2. Balancing
3. Waiting 5 sec when balanced out
4. Driving off the seesaw

The car could be started from an angle up to 45° relative to the seesaw and detects the black line automatically. This took additional time up to 5 seconds. After leaving the seesaw the car keeps moving for ~5 seconds and stops automatically.

It also had a separate balance mode where the car only balances the seesaw continuously.

Final Result:

Every requirement could be implemented and works like intended.

b) What can this project be used for?

Showcasing/Exhibition:

The car is very reliable and could be used for demonstrations in front of an audience. It has an interesting sequence of events that can be manipulated by an audience (laying weights on the seesaw and the car balances it). It turned out that the batteries last very long and it should be possible for the car to keep re balancing for several hours.

Technical demonstration for students:

The car combines a lot of basic principles of robotics, sensor technology, bus systems, control algorithms and more. The different control modes could easily demonstrate the difference between P, PD and PID control systems. The angle measurement includes different communication systems with electrical elements, filter technology and the digital implementation of sensors in general. The line tracking describes a simple way to navigate robots in their surroundings. Also the hole code is written in C program language and could be used as an example of how the language is used in robots.