HACETTEPE UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ELE 389-390

INTERNSHIP REPORT

ATILLA KÜRŞAT AKÇAY

21828092

PERFORMED AT

FOTONİKS ASKERİ ELEKTRONİK A.Ş.

15/08/22 - 26/09/22

30 WORK DAYS

STAMP AND SIGNATURE OF WORKPLACE

CONTENTS

**Organizational chart can be found in next page.

Introduction

I did my internship at Fotoniks Askeri Elektronik A.Ş. Department I worked at was research and development. Main focus of this company is, as the name suggests, photonics field of electronic devices. My motivation for choosing this place was that I want to work in defense industry in my future career, and Fotoniks is a rising defense contractor in Turkey.

During my internship, my work focused on embedded systems. My motivation for embedded systems is that I'm talented at programming and I love working with electronic devices, and with these two combined, it was reasonable for me to work in embedded devices field. As this is a defense ministry firm, the project engineers are working on is classified information. I helped and watched engineers with every opportunity I got and learned lots of valuable stuff from watching and collaborating with them. My biggest contribution to the project they are working on is that I wrote a program to help them test their device. Then I worked with lead engineer on testing the device.

Company Information

Fotoniks A.Ş. was founded in 2010 by Cem Yazıcıoğlu. During this brief time company grew exponentially and is continuing its growth. Address of workplace is; İlkbahar Mahallesi 571. Cadde 607. Sokak, D: No:11, 06550 Çankaya/Ankara

About my department

My department was research and department. Department organizational chart can be found in previous page. My position on this team was embedded systems intern. Main topic of this department is research and development of the systems Fotoniks A.Ş. creates.

About hardware and software systems

I mainly worked on STM32 microcontrollers. I was assigned 2 Arduinos, STM32G431, STM32F407, STM32L452, WS2812 LED strip, DP83848 external PHY and loads of components which I could freely use such as LEDs, sensors, resistors, UART to USB adapters and more. Anything I requested was supplied to me during my internship. I used Visual Studio Code, STM32CubeIDE and Arduino IDE during my studies as software systems.

Technical Presentation of the Workplace

Fotoniks, as the name suggests, is a company that specialize in photonics field of electronic devices. As this firm is a defense industry firm, my workplace didn't allow me to share any pictures or include any sensitive information that's not available to public due to my non-disclosure agreement. I am only allowed to use information available on the website. I will explain these solutions as explicitly as I can. Organizational chart can be found in previous page. Company told me they can't write names on it because it is classified.

Products Made by Fotoniks

Single Personnel Products

HNA: HNA (Holografik Nişan Aleti) is a product created by FOTONİKS A.Ş. and used by Turkish Army that is compatible with magnified optics and night vision systems. It uses a laser hologram technology. Technical specs are as follows.

Lenght	≤96.5 mm
Width	≤58.4 mm
Height	≤73.7 mm
Weight (w/out battery)	≤320 grams
Water Resistancy	10m
Battery Life	1000 h (mid-level brightness)
Rail Interface	MIL-STD 1913
Brightness	20 Level @ Day Mode 10 Level @ NV Mode
Laser Hologram Reticle	68 MOA Circle 1 MOA Aiming Dot

ATAK: ATAK (Ağ Tabanlı Askeri Kamera) is a product compatible with ballistic helmets to record and broadcast operations to headquarters. Technical specs are as follows.

Detector	5 MP CMOS
FOV	113°
Resolution	1920 X 1080 Piksel
Broadcast	4G/3G, Wi-Fi, Bluetooth
Video	H.265 ve H.264
Positioning	GPS
Interface	USB Type-C
LED	IR or Visible
Memmory	128 GB
Battery	2700 mAh
Battery Life	14 h
Operational Temperatures	-32°C +55°C
Environmental	IP67

TAG3B: TAG3B (3 Büyütmeli Dürbün) is a product that has capabilities of magnifying image 3X and can be used with holographic and red dot sights. User can fine tune the image with adjustment turrets. Technical Specs are as follows.

Magnification	3X
Dioptry	Yes
Objective Diameter	30 mm
FOV	7.0° (±0.05°)
Lens Coating	AR Coated
Eye Relief	65mm ± 5mm
Length	≤110 mm
Weight	≤340 gr
Environmental	IP67
Operating Temperatures	-32 °C to +55 °C

B-G³: B-G³ (Binoküler Gece Görüş Gözlüğü) is a high-performance military grade night vision system that can be plugged to modern ballistic helmets. Technical specs are as follows.

I2 Tube	Gen3
Lens	F1.18, 22.5mm
Eye Relief	30mm
Exit Puppil	8mm
Magnification	1X
FOV	40º
Battery Life	40 h (1 x CR123)
*Battery Life (PACK)	100 h (4 x CR123)

Tag-MR: Tag-MR (Mini Refleks Nişangah) is a quick acquisition sight that maintains fast and accurate shooting to the weapon. Technical specs are as follows.

Magnification	1X
Parallax	+/- 1 MOA @ 91,44 metre
Objective Diameter	24 mm x 17mm
Illumination Levels	0-10
Lens Coating	Anti Reflective Coating
Auto Turn off	4 h
Eye Relief	Unlimited
Zeroing Adjustment Range	90 MOA
Battery Life	≥2000h (mid brightness)
Size	48 x 27 x 24,5 mm
Weight	
Environmental	IP67
Water Resistance	30 min@1m
Operational Tempertures	-30 °C to +50 °C

OD – 60: OD – 60 is a device that helps in aerial support scenarios that can be used in planes and helicopters. It detects friendlies in these scenarios and helps troops immensely. It has a 2-degree infrared flash that emits a light which cannot be seen with eyes and can emit a green light to show the aerial vehicles position to friendlies. It contains 2 visible spectrum and 4 infrared LEDs.

Wavelength	Visible (515 – 525 nm)
	IR (830 – 850 nm)
IR Mode	Low (1 x LED – 20mW)
	High (3 x LED-90mW)
Body	Nylon 6/66 Polyamid
Cover	UV Protected Polycarbonathe Polymer
Environmental	IPX8
Drop Resistance	2 m
Weight	≤51 grams
Battery	1 X 3V – CR123

GÖKBÖRÜ: GÖKBÖRÜ is a military grade tactical weapon flashlight which operate at both IR and visible wavelengths. Technical specs are as follows.

Length	≤ 100 mm
Weight	≤ 130 g
Battery Life	White Light: 60 min.
	IR: 180 min.
Environmental	IPX7
IR Wavelength	830-860 nm
LED	White Light/6300°K
Lens	Borofloat Tempered - AR Coated
Drop Resistance	2.00 m
Power	White Light: 450 lumens (Optional 550 lumens)
	IR: 200mW (Optional 250mW)
Body	Aerial Grade 6061 Aluminum
Surface Coating	MIL-SPEC Hard Anodized Type III – Matte Black
Rail Interface	MIL-STD 1913

BÖRÜ: BÖRÜ is a military grade tactical weapon flashlight that emits visible wavelength light. Technical specs are as follows.

Length	
Weight	≤100 g (w/out battery)
Battery Life	60min.
Environmental	IPX7
LED	6300°K
Lens	Borofloat Tempered AR Coated
Drop Resistance	2m
Power	500-550 lumens
Body	Aerial Grade 6061 Aluminium
Surface Coating	MIL-SPEC Hard Anodize Type III – Matte Black
Rail Interface	MIL-STD 1913
Tape Switch	Optional

OD-250K STROBE LIGHT: OD-250K is a strobe light that helps with aerial identification of ground troops. It operates in infrared, visible spectrums and has a unidirectional mode. Tech specs are as follows.

Size	≤11,9cm x ≤5,9cm x ≤3cm
Weight	≤ 120 grams
Battery	2 X AA Lityum
Battery Life	40 h
Strobe Rate	50±10 strobe
Water Resistancy	10 m
Visbility	>9 km

Platform Integrated Systems

M-27 PERISCOPE: M27 PERISCOPE is a single glass monoblock prism that eliminates disadvantages of acrylic periscopes. It provides high resolution image. It has a defrosting circuit and a laser protection filter. Technical specs are as follows.

Optics	1:1 / Mono Block Galss
Laser Protection	OD6 @1064nm
FOV (Horizontal)	110° (Yatay)
Defrost	450 W/m2, 24VDC
Electrical Interface	MIL-C-26482 C8-33PW
Vibration Resistance	1,5G@60...400Hz
	1G@400...2000Hz
Shock Resistance	10G 6ms
	500G 0,5ms
Water Resistance	1,5 bar
Weight	

AVCI: AVCI is a thermal camera designed to enhance driver's vision and provide safe driving at nighttime and low visibility conditions. Tech specs are as follows.

Detector	Uncooled - Microbolometer
Detector Resolution	384 x 288
FOV (Narrow/Wide)	42°/55°
NETD	
Pixel Size	17µm
Wavelength	8-14µm
*Detection (Human-1.8m x 0.5m)	200m
*Detection (Vehicle-2.3m x 2.3m)	600m
Size	58.5mm x 58.5mm x 120mm
Weight	600g
Connector	RS 232 Serial Port + Video + KPSE01E12 - 10PDZ Power Connector

VNIR: VNIR Driver Vision Camera provides enhanced depth perception video to driver and the driver can see obstacles on the road in low visibility conditions with this device. Technical specs are as follows.

Detector	CMOS
Detector Resolution	640x480
FOV (Horizontal)	44°
Pixel Size	3.75 µm
Wavelength	400 – 1100 nm
Size	96x60x60 mm
Weight	
Communication Interface	RS422, Ethernet
Video Out	Analog PAL, AHD, Ethernet
Operating Temperatures	-32 °C - +55 °C

KUZGUN: KUZGUN is a thermal camera that supplies the vehicle it is connected to a 120^{o} camera system. This technology helps with vehicle perimeter safety and situational awareness is improved. Technical specs are as follows.

Detector	Uncooled- VOX
Detector Resolution	640 x 512
FOV (Horizontal)	120°
Pixel Size	17µm
Wavelength	8-12µm
Size	280mm x 120mm x 220mm
Weight	
Communication Interface	Ethernet,(RS 422 Optional)
Video Out	Analog,Ethernet (H.264)
Operational Temperatures	-32°C /+55°C

DESERT CAT: DESERT CAT is a thermal camera designed to enhance driver's vision and provide safety at nighttime and low visibility conditions. It has a manual pan and tilt system. Technical specs are as follows.

Detector	Uncooled - VOX
Detector Resolution	640 x 512
FOV (Narrow / Wide)	37.02°/45.4°
NETD	≤45mK
Pixel Size	17µm
Wavelength	8-14µm
Video Out	PAL
*Detection (Human- 1.8m x 0.5m)	450m
*Detection (Vehicle - 2.3m x 2.3m)	1000m
Power	28V DC Nominal (18-36VDC)
Weight	
Image Control Module	Optional

MARTI: MARTI is a thermal camera system that provides day and night vision for marine platforms. Technical specs are as follows.

Thermal Detector	Uncooled- VOX
Thermal Detector Resolution	384x288 (640X480 optional)
FOV (Narrow/Wide)	11,19°/14,88°
NETD	
Pixel Size	17 µm
Wavelength	8 µm – 14 µm
Video Out	Analog (PAL)
Human Detection (50cmx180cm)	900m
Vehicle Detection (230cmx230cm)	2000m
Day Camera	43X

DAT: DAT is a fusion driver camera that combines two images from different wavelengths to provide an image that can improve depth perception and situational awareness for the driver. Technical specs are as follows.

Thermal Detector	Uncooled-FPA
Thermal Detector Resolution	384 x 288
Thermal Pixel Size	17 µm
Thermal Wavelength	8 µm-14 µm
NETD	
Frame Rate	50fps
Low Light Detector	CMOS
FOV	36°x27°
Environmental	IP67
Operational Temperatures	-32°C/+55°C
*Detection (Human-1.8m x 0.5m)	300m
*Detection (Vehicle-2.3m x 2.3m)	900m

KAPLAN: KAPLAN is a thermal camera designed to enhance drivers' vision and provide safe driving at night and low visibility conditions. Technical specs are as follows.

Detector	Uncooled- VOx
Detector Resolution	640x512
FOV	45° x 37°
NETD	50 mK
Pixel Size	17 µm
Wavelength	7-14 µm
*Detection (Human-1.8m x 0.5m)	360 m
*Detection (Vehicle-2.3m x 2.3m)	880 m
Size	96 x 60 x 60 mm
Weight	
Connector	D38999/24WD35PN

UNIT LEVEL SYSTEMS

ATLAS-2A: ATLAS-2A is a surveillance and reconnaissance system that can detect, recognize and identify threats day and night in adverse weather conditions. It can range, coordinate, point, mark and track targets. Technical specs are as follows.

Thermal Detector	Cooled MCT
Thermal Detector Resolution	640x512
FOV (Narrow/Wide)	1,9°/12,5°
E-Zoom	2X
Wavelength	3 µm – 5 µm
Video Out	Digital-Ethernet-HD-SDI-Analog-CCIR
Human Detection (50cmx180cm)	13000m
Vehicle detection (230cmx230cm)	20000m
Laser Range Finder - Range	≤20000m
Laser Range Finder - Wavelength	1,54µm
GPS Accuracy	2,5m
GPS Data Type	UTM/Geo
DMC Accuracy	1,5°(+/-0,5°rms)
Laser Pointer	≥2000m
Size	750mm x 330mm x 400mm
Weight	
Operational Temperatures	-32 °C - +55 °C

ATLAS-2B: ATLAS-2B is the enhanced version of ATLAS-2A that has mostly same capabilities albeit some improvements. Technical specs are as follows.

Thermal Detector	Uncooled- VOX
Thermal Detector Resolution	640x512
FOV (Narrow/Wide)	2,77°/24,5°
E-Zoom	4X
Optical Magnification	9X
Wavelength	8 µm – 14 µm
Video Out	Analog (PAL)- Dijital (Camera link 600bit/s)
Human Detection (50cmx180cm)	5000m
Vehicle Detection (230cmx230cm)	10000m
Day Camera	36X
Size	838mm x 400mm x 500mm
Weight	
Operational Temperatures	-32 °C - +55 °C

SWIR: SWIR Surveillance System is a durable digital VIS-SWIR camera that offer high precision imaging. With the help of its sensor size camera offers high resolution VIS-SWIR image. Technical specs are as follows.

SWIR Detector	InGaAs Photodiode
SWIR Detector Resolution	640x512
Pixel Size	15 µm
Wavelength	0.4 µm – 1.7 µm
Human Detection (100mm lens)	4000m
Vehicle Detection (100mm lens)	10000m
Lens	100mm (Optional 300mm Lens)
Video Out	PAL
Day Camera Detector	1/2.8-type Exmor R CMOS
Day Camera FOV (Narrow/Wide)	2,3°-63,7°
Day Camera E-Zoom	12X

LCS: LCS (Laser Communication System) offers fast and secure wireless data transmission unaffected by jammers. Technical specs are as follows

Laser Wavelength	1550nm
Range	≤5000m
Speed	1000 Mbps
Environmental	IP67
Power Consumption	50 W
Laser Class	Class 1M
Band Liciense	Not Required
Alignment	Telescope
Stabilization	Auto

TOPUZ: TOPUZ is designed for perimeter surveillance of military outposts and police stations. It provides thermal and visible light imaging with protection against small projectiles with its armor. Technical specs are as follows.

Thermal Detector	Uncooled- Microbolometer
Thermal Detector Resolution	640x480
FOV (Narrow/Wide)	30°/40°
Pixel Size	17 µm
E-Zoom	8X
Wavelength	8 µm – 14 µm
Human Detection (50cmx180cm)	500m
Vehicle Detection (230cmx230cm)	1300m
Day Camera Detector	CCD
Day Camera Wavelength	400-700nm
Day Camera FOV (Narrow/Wide)	1,5°/57°

I can't give any other info about my workplace as it is a defense industry firm and data security is a big concern. I signed a non-disclosure agreement and when I asked my mentor, he told me I can only put information available to public in my report.

Information About My Supervisor

Musa ÜZÜM – Lead Electronics Engineer - 0507 441 13 51

1st Week:

15/08/22 Monday

In my 1st day I got introduced to embedded systems, learned about Arduino IDE and some capabilities of Arduino cards. I programmed several LEDs to pulse with a timer then I put a button to light up a LED. Meanwhile I installed STM32CubeIDE for my future projects and learned about this IDE too.

File(s) I worked on:

https://github.com/atilla-kursat/Arduino-Projects/tree/master/Karasimsek

https://github.com/atilla-kursat/Arduino-Projects/tree/master/button

16/08/22 Tuesday

In my 2nd day, I learned analog signal processing and PwM (Pulse with Modulation) capabilities of Arduino Uno. I created a project which makes relative LEDs blink when joystick is in their direction. Then I created a small project where I use a supersonic sensor as a proximity sensor. This sensor emits a supersonic sound which human ear can't hear, and it can also check if a same frequency sound is present with a sensor. Distance is found by using the speed of sound in the air. When the sound emitted reflects from a surface it can be detected by the sensor and the time this operation happens is recorded by microcontroller. Then distance can be easily found by multiplying the speed of sound with the travel time. Something to note is that result should be divided by 2 as sensor can only detect refracted sound waves and it needs to travel to surface and back. Thus, we easily find the distance and depending on the distance a LED will light up. Started working with STM32L452 and learned about HAL libraries. I recreated the setups I have used in Arduino with STM32L452. Digital operations were successful but analog signals were not recognized by the microcontroller. I tried to figure out what was wrong with my code and asked my mentor about it. He checked my code and setup then concluded that microcontroller was faulty. He told me to wait a day for a new microcontroller.

File(s) I worked on:

https://github.com/atilla-kursat/Arduino-Projects/tree/master/Proximity_Sensor

https://github.com/atilla-kursat/STM/tree/master/TEST2

https://github.com/atilla-kursat/Arduino-Projects/tree/master/Joystick

17/08/22 Wednesday

In my 3rd day, I worked with an Arduino again and printed several messages to a 16x2 LCD screen. Programming this screen was easy as Arduino has needed libraries already baked in. Real challenge was managing cables as the LCD screen had sixteen pin connectors. I connected it to a breadboard and worked on it. After that I created the classic game

“snake” on the Arduino. Later that day my mentor gave me a new card, STM32G431KB. I started working on this card and used the LCD from my previous project again to write messages. This time it was more difficult as libraries weren’t included in the STM32CubeIDE. I imported drivers for this task to my project and integrated it to my code, and successfully made it work.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/lcd

https://github.com/atilla-kursat/Arduino-Projects/tree/master/Simple_Snake_Game

18/08/22 Thursday

In my 4th day, I worked with the joystick to understand ADC capabilities of the microcontroller. However, joystick signal was being wrongfully read by the microcontroller, so I connected a potentiometer to find just exactly how much resistance I needed for correct results. Then I connected a 390 Ω resistor in series, after that signal was correctly read by the microcontroller. I studied the PwM capabilities of the microcontroller and used a LED and changed its duty cycle to dim and brighten the LED. Then my mentor and I had a talk about an engineer’s duties, and he lectured me about what people expect from an electrical and electronics engineer, how to become a better embedded software developer, how I can improve myself upon these fields. He showed me how to find relevant datasheets of the card I’m working on and tasked me with studying the datasheets of the STM32G431KB which was approximately 2400 pages long.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/adctest

19/08/22 Friday

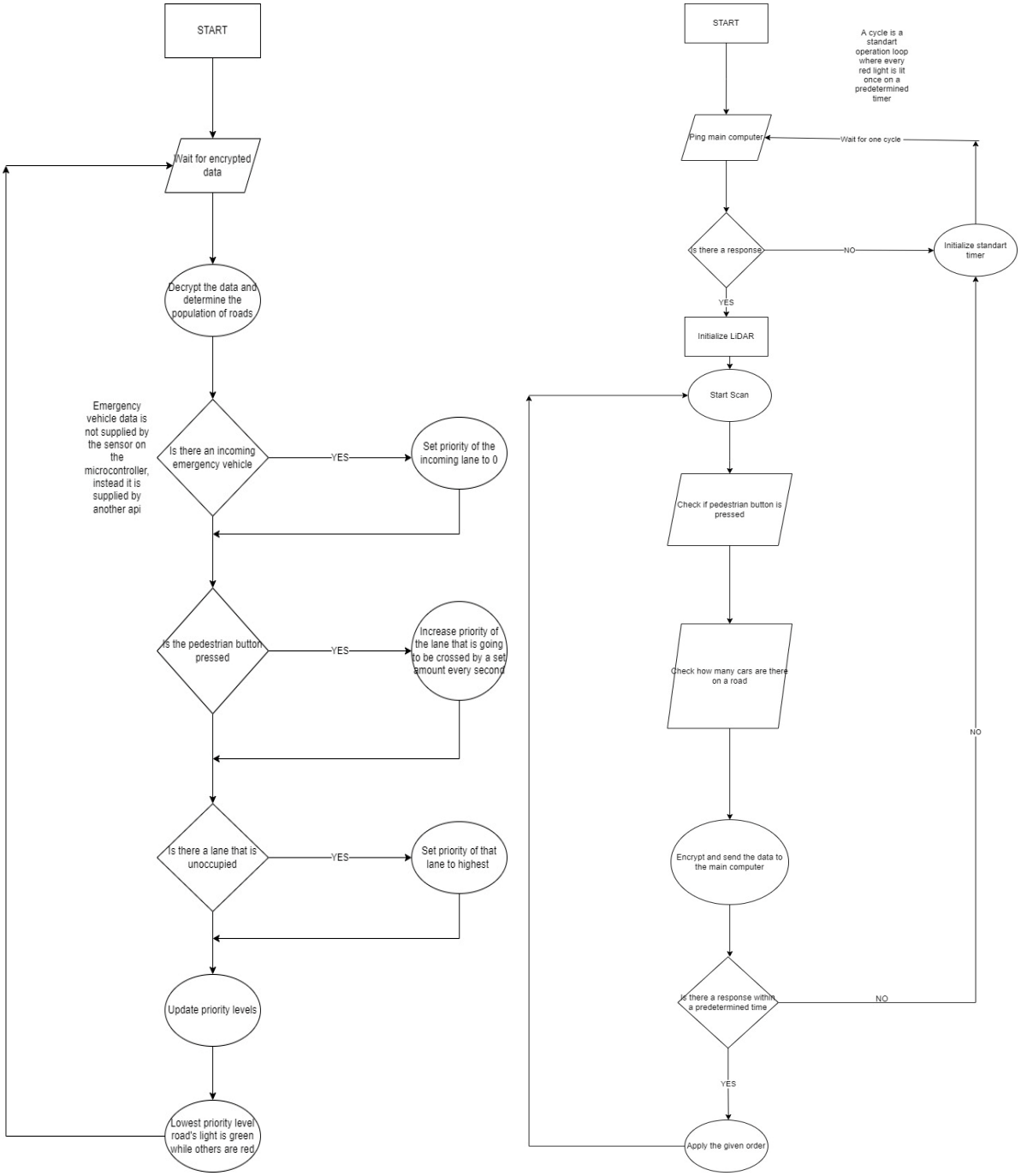
In my 5th day, I continued to study the datasheets. I studied approximately 800 pages and continued studying throughout the weekend. Here is a small list of the things I learned.

1- General architecture and working conditions of the microcontroller.
2- Modules the microcontroller possess and capabilities of said modules.
3- Exploiting these modules for faster processing and calculation of data.
4- What is DMA (Direct Memory Access) and how to use it.
5- What is CORDIC and how to use it.
6- How to work with FMAC.

2nd Week:

22/08/22 Monday

In my 6th day, I continued to study the datasheets in the morning. After I studied about 300 pages, my mentor tasked me with finding a project with no restraints to teach me the general approach to electronics engineering. I created a flowchart for a project through the day and presented it to my mentor. Project was a traffic control system using LiDAR sensors and depended on a main computer for decision making parts of the project.



After my presentation we discussed the project and after that he said he will give me a project to work on.

23/08/22 Tuesday

In my 7th day, I was tasked with creating an API (Application Programming Interface) to control a WS2812 LED strip with 8 programmable LEDs. Expectations from the project were:

1-Main computer and microcontroller should communicate over serial port using UART protocol.

2-Each LED should be programmable separately.

3-There should be data corruption checking mechanisms.

4-Data corruption check mechanism on the microcontroller side should use CRC (Cyclic Redundancy Check) hardware module of the microcontroller.

With these parameters I started working on the project. I read the datasheet of WS2812 and researched how it worked.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/led_strip

24/08/22 Tuesday

In my 8th day, I programmed the WS2812 driver on the microcontroller. Clock speed for the microcontroller was selected separately to synchronize with the device which uses an 800 kHz controller. Microcontroller itself is 170 MHz and clock speed for DMA module, which transmits the data to the WS2812, was selected as 72 MHz with prescaler selected as 90. With this setup, 800 kHz clock speed of the LEDs are matched as 72 MHz/90 = 800 kHz. After that, by the datasheet of the WS2812, for an incoming '1' bit duty cycle should be 68%, and for an incoming '0' bit it should be 32%. Using this information, I successfully programmed a driver to the WS2812 which can select color and brightness for each LED. Then I started working on the API.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/ledstrip.cpp

https://github.com/atilla-kursat/STM/tree/master/led_strip

25/08/22 Wednesday

In my 9th day, I worked on the API. I learned how to use win32 calls to communicate with the microcontroller. Using the "windows.h" library, I successfully created a barebones API which could control LEDs on the WS2812. After confirming communication and application of the signal sent by the computer, I started to study the CRC (Cyclic Redundancy Check) mechanism of the microcontroller. STM32G431KB has a hardware solution for CRC to make it faster and it is much more efficient than the software solution. To match the result generated on the microcontroller, I found the generator key and tried to start to implement a software solution for the main computer.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/ledstrip.cpp

https://github.com/atilla-kursat/C-repos/blob/master/crctest.cpp

26/08/22 Thursday

In my 10th day, I decided to create the necessary functions for the CRC32 calculations myself. The problem was that doing bitwise XOR calculations on C was harder than I imagined. Using the generator key, algorithm starts from MSB (most significant bit) and aligns the MSB of the generator key according to it. If the bit is '0' a leftshift operation is made on the original string of bits. When the algorithm encounters a '1' it XOR's the generator key with original string. Until end of original string is reached this task goes on. CRC generator polynomial for this task was Ethernet standard 0x104C11DB7. However, it was a task that was proven to be too hard for me, as I couldn't find an efficient solution for it by myself. Then I decided to use a "zlib.h" library which has a CRC32 solution as a function, but this time data types I selected for this task was not compatible with this function. I converted the types to needed types. I spent the entire day working on it.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/crctest2.cpp

3rd Week

29/08/22 Monday

In my 11th day, I continued to work on the API and designed packets to be read by the microcontroller. Much of the work went into CRC32 as the result on the computer and microcontroller didn't match. After working on it for a while I found the solution and successfully implemented fault checking mechanisms. The problem was zlib's solution of crc32 XOR'd the solution with 0xFFFFFFFF. I corrected it with doing the same operation on the microcontroller as well.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/ledstrip.cpp

https://github.com/atilla-kursat/STM/tree/master/led_strip

30/08/22 Tuesday ---------- Victory Day

31/08/22 Wednesday

In my 12th day I fixed some bugs and finished my project. I presented it to my mentor. After that I started working on my next project which is to create a USB HiD device. I started my research on HiD devices and the architecture of USB. Requirements for this project were:

1- A keyboard should connect to a microcontroller and recognize keystrokes.
2- Recognized keystrokes should be sent to a main computer and be accepted as valid keyboard inputs.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/ledstrip.cpp

https://github.com/atilla-kursat/STM/tree/master/led_strip

01/09/22 Thursday

In my 13th day I continued my research on the topic. I read topics from usb.org and Microsoft.com and began working on the report organization. Reports are the main communication principle behind HiD devices. Each report type has its own organization, and I should program my microcontroller according to these. After my research I also realized that my microcontroller STM32G431KB does not met the requirements for this project as it does not have a VBUS which delivers power to attached keyboard. I told my mentor about it, and I was told I would be assigned a new card.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/HiD%20device

https://github.com/atilla-kursat/STM/tree/master/custom_hid

02/09/22 Friday

In my 14th day I was assigned my new card STM32F407VGT. This board had a small issue where its programmer module was desoldered thus I needed another way to flash my programs to this board. I began my research on how to do it and found out that ST-LINK can be used in this manner. However, for a ST-LINK connection I needed another board that had a programmer, and my very first STM32 card STM32L452 was suitable for this task. I connected necessary connections and successfully initiated and programmed STM32F407. After that I continued working on my project and wrote a barebones program. Sadly, it didn't work, and I continued to work on it.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/new_board_hid

4th Week:

05/09/22 Monday

In my 15th day I continued my work on the project and I gained better understanding on the subject. I understood how a package is formatted and what data it carries and made some changes on my project. I learned that I cannot use a microcontroller as a host and a device simultaneously and I needed another microcontroller for this task. Luckily my STM32G431KB was suitable as a device because I can fetch keyboard data to my host card STM32F407VGT and transfer it to STM32G431KB using UART. Then, the transmitted data will be reformed in a format that is recognized as a USB device by the computer. I needed 3 microcontrollers for the whole task:

1- STM32F407VGT as the host, keyboard connects to this microcontroller
2- STM32L452 as the programmer of STM32F407VGT using ST-LINK
3- STM32G431KB as the recognized HiD device by the computer

I started programming said cards.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/new_board_hid

https://github.com/atilla-kursat/STM/tree/master/hid_i2c_device

06/09/22 Tuesday

In my 16th day I programmed STM32F407VGT as a host to a keyboard. Then I programmed STMG431KB as a device to the computer. Then I programmed the communication channels and sent my data on the STM32F407VGT through UART. After the data is received it is formatted into the keyboard struct. Then it is sent through the USB channel. However due to hardware limitations, I couldn't test this project. I couldn't get my hands on a OTG cable and transferring data through pins on the MCU could potentially damage the computer or the MCU and my mentor advised against it. Then my mentor told me to create another project where I get analog input from a joystick and create a driver for it to be recognized by a windows computer. I started working on it.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/new_board_hid

https://github.com/atilla-kursat/STM/tree/master/hid_i2c_device

https://github.com/atilla-kursat/STM/tree/master/joystick_hid_device

07/09/22 Wednesday

In my 17th day, I continued to work on the joystick driver. I had a headache so I couldn't focus too much on the project. I encountered some problems on the ADC module of the STM32F407VGT. There was too much noise, and I couldn't get a good reading on the joystick. I programmed a noise reduction algorithm that averaged 100 readings to get a good result but nonetheless readings were wrong. Then I tested if the joystick was faulty by connecting it to another MCU STM32G431KB. I needed separate channels for separate axis. My readings for one channel were correct but the other channel was not correctly reading the joystick. I switched connections to check if the joystick is faulty, but it wasn't the case as this time same channel was getting wrong readings. I theorized that by enabling UART pins that was close to the channel that was getting wrong readings, I created a noise source for the ADC. I needed another way of solving this and I solved it by using an Arduino. My final setup war Arduino gathering ADC data, mapping it between -127 and 127, then sending this data through UART to the MCU, STM32F407VGT. Then incoming data is checked for any corruptions and timing issues, and if it is correct, it is sent over to the windows computer. However, my program didn't work at first.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/joystick_hid_2

https://github.com/atilla-kursat/STM/tree/master/Joystick_hid_3

https://github.com/atilla-kursat/Arduino-Projects/tree/master/UART_ANALOG_READ_AND_SEND

08/09/22 Thursday

In my 18th day I took another look at the program with a clear mind and easily found the reasons the code wasn't working. Problem was that I modified the report description but forgot to change the report description size. I fixed the bugs and completed the project. I then presented the project to my mentor. We then talked about what was next for me and he asked me to decide what do I want to learn. I wanted to create a project with ethernet as

the communication channel. He told me to research lwIP and ethernet communications. I learned about the terminology. Later in the day my mentor gave me the external PHY connector, DP83848. This module allows me to connect an ethernet cable to my microcontroller. He told me to always use static foot band while using it and always store it in a ESD bag both of which I didn't use before. I did know that static electric discharge can potentially damage or lower their life expectancy, but I didn't realize it was this big of a deal.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/joystick_hid_4

09/09/22 Friday

In my 19th day I had dived deeper into lwIP and ethernet communication protocols. I learned about how devices connect to each other via ethernet and how IP address works, how MAC address works, how devices send packets, what these packets include and how a packet loss occurs, how to counteract corruption on the lines and how lwIP is implemented on the STM libraries. I studied extensively on this subject and read the manual of DP83848, lwIP application note and ethernet connection part of the STM32F407 manual. Then I connected the DP83848 to my MCU and programmed the card to respond to a ping. However due to reasons unbeknown to me it did not work. I continued my study about it. Later in the day team leader of the electronics division asked me to do a small project and deliver it to him by Monday. It was a small UART testing program to a project I cannot write about in here due to my non-disclosure agreement with the firm.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/lwip_example

https://github.com/atilla-kursat/STM/tree/master/Ethernet

https://github.com/atilla-kursat/STM/tree/master/uart_transmit_test

https://github.com/atilla-kursat/STM/tree/master/UART_test

5th Week:

12/09/22 Monday

In my 20th day, I continued to work on ethernet connection and troubleshooting what causes my computer to not discover the device. I had a better understanding on the subject and learned a lot about lwIP. Sadly, my attempts on the project were futile as it didn't work properly. I suspected that something hardware related was holding me back, but I couldn't find what it was. I checked device connections, settings on my microcontroller, if the pin was faulty or jumper connections. During my troubleshooting I was asked to do code another small UART test program for the project I can't talk about due to my non-disclosure agreement with the firm. So, I spend the rest of the day working on it.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet2

https://github.com/atilla-kursat/STM/tree/master/ethernet3

https://github.com/atilla-kursat/STM/tree/master/UART_test

13/09/22 Tuesday

In my 21st day, I completed the test program and started to reassemble my ethernet project. I still couldn't find the problem at first and asked my mentor for help. He checked my code and told me to check my connections again to see if it works properly. After testing each jumper separately, I saw that one of my jumpers was sometimes not conducting, and it was the cause of my problems. Then I presented it to my mentor, and he told me to create a project where microcontroller discovers and pings all devices on a network. I started working on it. I learned about OSI layers, how a ping works, what it contains, protocols that make ethernet communication possible, differences between UDP and TCP and which should be used for which application.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet3

14/09/22 Wednesday

In my 22nd day I continued my work on network announce project. As a prototype I created a program where microcontroller can communicate with another device. I configured the microcontroller both as a client and a host. It worked flawlessly but as I learned more about it, I realized I cannot use UDP for my project. I needed to learn about ICMP to ping a device and get a response. However, lwIP libraries only contain a ping response function, you cannot send a ping using only lwIP libraries. I started programming a ping function. To program a send ping function, I needed to port the needed libraries to my version of lwIP. This was not an easy task however, as every time I ported a .h file another thing broke in the code. I continued to work on it.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet_icmp

15/19/22 Thursday

In my 23rd day I continued to work on my project. Porting a ping program from scratch was hard, as ICMP packages isn't configured for sending on lwIP. Porting several .h files and making sure they worked okay and configuring a ping packet consumed most of my day. I needed to learn what a ping packet contains and what response it generates. Then I needed to code a struct that contained this data and send it via RAW API. Computer recognized it as a ping from the microcontroller, so it was successful.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet_icmp

16/19/22 Friday

In my 24th day I continued to work on my project. However, there was a problem with the results. Pings were acknowledged by the computer, but it didn't respond to pings. I researched the root of this problem. I changed firewall settings, tried pinging with another ID, tried to ping with and without checksum but it didn't budge. Then I tried the same thing on Linux, and it worked. It was a problem with Windows 11 that I couldn't resolve. Then I cleaned up the code. I worked on this problem entire day.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet_icmp

6th Week

19/09/22 Monday

In my 25th day I finally started on my API. I learned about "Winsocket2.h" library and its functions. I implemented a ping function, a function that makes MCU ping the computer, a function that demands a report from the MCU, a function that accepts an IP address as input and makes MCU ping that IP address, a function that accepts an IP address as input and makes MCU send a report to that IP address. On the MCU side I programmed operation codes that makes communication meaningful between computer and MCU. I faced some problems about how Windows handles sockets. There were lots of documentation about how to do it within Linux as socket creation and handling is a lot easier on Linux, but documentation for windows was poor. I read some documentation and that was it for the day.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/ethernet_api

https://github.com/atilla-kursat/C-repos/blob/master/iparraytest.cpp

https://github.com/atilla-kursat/C-repos/blob/master/pingAPI.cpp

20/09/22 Tuesday

In my 26th day, I continued working on my API and I finished it. Then I presented the project to my mentor. He told me there were no problems with functionality, but code was not clean. He told me to learn clean coding principles because as my projects gets bigger, I need to have a portable and understandable code, not just functional. I studied how to write clean and portable code most of the day. Towards the end lead engineer asked me to flash a testing program to a microcontroller to test the project they are currently working on, which I can't talk about.

File(s) I worked on: https://github.com/atilla-kursat/C-repos/blob/master/pingAPI.cpp

https://github.com/atilla-kursat/STM/tree/master/ethernet_api

https://github.com/atilla-kursat/STM/tree/master/UART_test

21/09/22 Wednesday

In my 27th day, I spend most of the day working on this test. I wrote several different variations of the UART code for troubleshooting and programmed another test program that tests ethernet capabilities of the card. I worked with the lead engineer in this, and it was a priceless experience for me to work on a real project. We tested what works and what doesn't and how we can make it work. I learned a lot about hardware side of the microcontrollers and PCBs. After that I started to assemble my internship report and spend the rest of the day with it.

File(s) I worked on: https://github.com/atilla-kursat/STM/tree/master/UART_test

https://github.com/atilla-kursat/STM/tree/master/ethernet_api

22/09/22 Thursday

In my 28th day I continued working on my internship report. Put links to all the relevant GitHub repos and formatted the document. I spend the rest of my time learning about clean code principles and inspecting neatly written codes.

23/09/22 Friday

In my 29th day I concluded my report. I gathered necessary information and put it into this report. I watched engineers working and learned valuable things and helped a little. This will be my final entry to this report as I will print it and deliver it to lead engineer by Monday to get it signed. My time here has been joyful, and I am glad I got the chance to do my internship in Fotoniks.

7th Week

26/09/22 Monday

Today was my last day here. It was also course registration day at Hacettepe so I selected my classes for August term. Then I talked with accounting and human resources about my paperwork. A new intern arrived today, and I showed her around the office. Then I helped hardware engineer most of the day on a project I can't write about because of my non-disclosure agreement. We tried different configurations for a circuit, and I learned about what to consider when doing a modification to a circuit that behaves unexpectedly. Then it was time for me to say goodbye to the people I have been working with for one and a half months. I bought baklava and gave it out personally and talked with them about how wonderful my experience was at Fotoniks. They all wished me good luck on my career. I organized all the things I've worked with and waited for my supervisor to sign my documents.

Performance And Outcomes

Applying Knowledge and Skills Learned at Hacettepe

To see all my schoolwork in practice, helped me love my future profession. I used information I learned in ELE225, ELE120, ELE336 and ELE107 to full extent during my internship. I also realized all the theoretical courses in our school helped me develop a critical thinking habit. I approached problems like an engineer and solved them like an engineer. I also benefitted from things I learned from our laboratory courses.

Solving Engineering Problems

One of the most important things I've learned here was how to read and get information from application notes. I became proficient in researching and applying setups. These combined with my previous knowledge helped me solve problems like an engineer. Rather than putting a band aid on my problems and going along with it I searched root cause of problems. I experienced which mistakes lead to which results and this helped me learn debugging in a much better way than before.

Teamwork

During most of my time spent here I was the only intern at this company. So, I did not have a project which I have done with my peers. However, this was advantageous for me because I improved myself without any constraints and I did teamwork with engineers themselves which improved me immensely in engineering solutions. To work with them on real projects and inspecting how they worked taught me lots of things I did not know about. All engineers were caring and three of them was invested in helping me.

During the times I was not the sole intern, I helped other interns in any way I can. I got to know four interns. I helped two of them on their projects and other one was not interested in working at all. I helped one of them with C++ and the other one with C. I also got help with embedded systems from one of the interns. Fourth intern came on my last day, so I just showed her around.

Multi-Disciplinary Work

As embedded systems requires good knowledge about software and hardware, I got lots of experience about both fields. I needed to study computer science standards and I learned a lot about hardware from documents. I also learned about photonics and mechanical part of the projects I can't talk about. There was a mechanical engineer that visited me often and showed me how they did designs.

Impact of Engineering Solutions

I learned how things designed here help our soldiers in battling fields, how it reduces casualties and what advantages it brings to our soldiers

Locating Sources and Self-Learning

Most important thing my mentor did that helped me was, he just pointed me at a direction and didn't hold my hand through my project. This way I got better at researching, at one point I was looking at Chinese websites to find why my setup was not working. My sources were mostly application notes of whatever I was using. My biggest resource was user manual of STM. I also benefitted a lot from resources Microsoft published on their website to program my APIs. USB.org was an immense help in my HID device project. lwIP application notes were lacking but helpful to understand the topic too. I also searched about other people's setups and how they dealt with problems.

Results and Comments:

During my time here, I can say that I enjoyed being here. Fotoniks is not as big of a firm compared to likes of ASELSAN, but I believe this company takes advantage of that. Projects are not blocked by corporate ladder and with the talent of the engineers working in this firm, things get done fast. Engineers were working tirelessly but by the way they do projects I can say they were ecstatic about it. They love their jobs and do long working hours voluntarily. My mentor was truly kind and understanding to me and helped me in every way he can. I helped the electronics lead engineer in a project and that work made me love my profession. To use my education in a real-world scenario gave me a reason to get out of bed in the morning. I also think one other advantage of not being a big firm is that employees are getting along better with each other well. I felt accepted in a small amount of time. During my time here engineers were in the testing stage of a project, and I had lots of chances to observe the process and I saw how they approached problems, how to get around inconveniences and deliver a project on time. Being a defense industry firm, there really is no place for mistakes. There were a lot of considerations made before attempting to change or fix something.

I studied embedded systems during my time here. Before my internship I thought about getting a microcontroller myself but never really had a chance to get one due to how expensive they are and to this day I postponed it. In my first day here, I got my hands on an Arduino and was fascinated by how much I could do with such simple programs. That day I decided that I want to continue working in this field and studied as much as I could. I couldn't bring MCUs to my house during weekends etc. but worked on them at night and tested the programs in the morning to compensate. I also learned about structures of Windows and Linux, practiced C++ and general computer science theory.

I was in the same room with technicians and learned a lot about handling electronic devices from them. I had a better understanding on oscillators, and I got better at soldering. I used a microscope to inspect devices I worked on, had a firsthand experience with some circuits.

Overall, I can say that my internship taught me a lot of invaluable things and I'm glad I did my internship in this firm. I didn't expect I would like this place this much and, in the end, I'm sad to leave here. I would recommend working in this firm to my classmates.

Appendix

All the files I have worked on, some photos and videos of my projects and digital format of this file can be found on my GitHub account: www.github.com/atilla-kursat

More information about Fotoniks A.Ş. can be found in www.fotoniks.com.tr