



MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS
ENGINEERING

EE300 Summer Practice Report

Student

Name : HALIL

Surname : TEMURTAŞ

Number : 2094522

Supervisor Engineer

ÖMER EREN CAN

KOÇULU

.. .. 2017

Contents

1	Introduction	2
2	Description of the Company	2
2.1	Company Name	2
2.2	Company Location	2
2.3	General Description of the Company	3
2.4	A Brief History of the Company	3
3	Orientation & Useful Programs	4
3.1	Pomodoro Technique & Pomotodo App	5
3.2	Database Structure & Airtable	6
3.3	Wiki Pages & Confluence Wiki	7
3.4	V-Model & Agile Methodology	8
3.5	Version Control with Git & Bitbucket	8
3.6	Microsoft Sharepoint	9
4	Python & Raspberry Pi	9
5	Solar Tracker System Project	13
6	MATLAB	13
7	Conclusion	13
8	References	13

1 Introduction

I have performed my summer practice in TÜRKSAT A.Ş. (Türksat Satellite Communications and Cable TV Operations Company - Türksat Uydu Haberleşme Kablo TV ve İşletme A.Ş). It is the sole communications satellite operator in Turkey. My internship lasted 20 days. Ömer Eren Koçulu, who is a mechatronics engineer in TURKSAT was our supervisor and he managed our internship program . . . In this report, I start with introducing the company. After that, I reported what I have performed and learned during SP. I have given detailed information about what I have done in projects: codes, simulations, circuitry, etc. At this part, I used many figures and photographs to better explain my works. After that, I summarized my report in “Conclusion” part. I included the sources of the documents from which I took help in the “References” part.

2 Description of the Company

Our circuit can be analysed mainly in five parts:

1. Company Name
2. Company Location
3. General Description of the Company
4. A Brief History of the Company

2.1 Company Name

TÜRKSAT A.Ş. (Türksat Satellite Communications and Cable TV Operations Company - Türksat Uydu Haberleşme Kablo TV ve İşletme A.Ş).

2.2 Company Location

Address-1: Ana Kampüs

Address-2: Gazi Teknokent

Phone: +

Fax: +

2.3 General Description of the Company

Türksat Satellite Communications and Cable TV Operations Company is the sole communications satellite operator in Turkey. It was established on 21 December 1990 as a state-owned company named Türksat Milli Haberleşme Uyduları (Türksat National Communications Satellites) in Gölbaşı, Ankara Province;[1] eventually incorporating the satellite services of Türk Telekomünikasyon A.Ş. and becoming Türksat A.Ş. on 22 July 2004.[1] Türksat A.Ş. also owns 100

2.4 A Brief History of the Company

1968 The Chief Engineering of Satellite Telecommunications Group was established within the General Directorate of PTT. 1968 Membership to the Intelsat (International Telecommunications Satellite Organization) Organization. The first telephone channels via satellite were co-established in the same year by taking the advantage of the ground stations in Yugoslavia and Iran. April 23rd, 1979 Our first satellite ground station AKA-1 (Ankara-1) was put into service by establishing 11 telephone channels with England. Thanks to commissioning AKA-1, communication with 13 countries was achieved by using the Atlantic-region satellites through Intelsat April 10th, 1985 The first fiber optic cable increasing channel capacity for communication was laid underground between Ankara (Ulus) and Gölbaşı Satellite Ground Station, so that the 140-Mb/sec. system was commissioned at 1310 nm wavelength. November 1985 AKA-2 ground station working with the Eutelsat system was commissioned. March 4th, 1987 The first video conference through satellite system in Europe was held in our country. December 21st, 1990 The "Türksat National Communication Satellites" contract was concluded with Aerospatiale Company from France. August 11th, 1994 Turkey's Türksat 1B satellite was launched and put successfully into 42° East orbit. October 10th, 1994 Türksat 1B was commissioned following the orbit tests. July 10th, 1996 Turkey's second satellite, Türksat 1C, was launched and put into 31.3° E orbit. November 7th, 1996 The founding agreement of Eurasiasat, the partners of which were Aerospatiale, manufacturing Türksat satellites, and Türk Telekom, was concluded. November 15th, 1996 The Company came into operation under the same Eurasiasat S.A.M. to be responsible for the manufacturing of Türksat 2A satellite and marketing its services. January 11th, 2001 Türksat 2A (Eurasiasat 1) satellite manufactured by Eurasiasat

company established in partnership with Türk Telekom and Alcatel company launched by Ariane 4 rocket from Kourou Base in South America. July 22nd, 2004 In order to conduct satellite communication services, which was previously conducted by Türk Telekomünikasyon A.Ş., under a new company, Türksat A.Ş. was founded by Law no. 5189. June 13th, 2008 Türksat 3A satellite launched from the French Guiana on June 13th, 2008 at 01:05 by Ariane 5 rocket and put into 42.0° East orbit. April 26th, 2013 Turksat 3USAT first domestic designed and manufactured low orbit satellite launched from Jinguan in China. February 14th, 2014 Turksat 4A communication satellite launched by Proton rocket from Baikonur Cosmodrome in Kazakhstan. October 16th, 2015 Turksat 4B communication satellite launched by Proton Breeze M vehicle from Baikonur Cosmodrome in Kazakhstan and put into 50° East orbit.

3 Orientation & Useful Programs

Throughout my internship, I have used several useful program recommended by our supervisor.

In this section, I will explain the techniques and programs that I used in my summer practice:

1. Pomodoro Technique & Pomotodo App
2. Database Structure & Airtable
3. Wiki Pages & Confluence Wiki
4. V-Model & Agile Methodology
5. Version Control with Git & Bitbucket

3.1 Pomodoro Technique & Pomotodo App

The Pomodoro Technique is a time management method developed by Francesco Cirillo in the late 1980s. The technique uses a timer to break down work into intervals, traditionally 25 minutes in length, separated by short breaks. These intervals are named pomodoros, the plural in English of the Italian word pomodoro (tomato), after the tomato-shaped kitchen timer that Cirillo used as a university student.

Closely related to concepts such as timeboxing and iterative and incremental development used in software design, the method has been adopted in pair programming contexts.

There are six steps in the technique:

1. Decide on the task to be done.
2. Set the pomodoro timer (traditionally to 25 minutes).
3. Work on the task until the timer rings.
4. After the timer rings put a checkmark on a piece of paper.
5. If you have fewer than four checkmarks, take a short break (3–5 minutes), then go to step 2.
6. After four pomodoros, take a longer break (15–30 minutes), reset your checkmark count to zero, then go to step 1.

The stages of planning, tracking, recording, processing and visualizing are fundamental to the technique. In the planning phase tasks are prioritized by recording them in a "To Do Today" list. This enables users to estimate the effort tasks require. As pomodoros are completed, they are recorded, adding to a sense of accomplishment and providing raw data for subsequent self-observation and improvement.

For the purposes of the technique, a pomodoro is the interval of time spent working.[1] After task completion, any time remaining in the pomodoro is devoted to overlearning. Regular breaks are taken, aiding assimilation. A short (3–5 minutes) rest separates consecutive pomodoros. Four pomodoros form a set. A longer (15–30 minute) rest is taken between sets.[1][6]

A goal of the technique is to reduce the impact of internal and external interruptions on focus and flow. A pomodoro is indivisible. When interrupted during a pomodoro, either the other activity must be recorded and

postponed (inform – negotiate – schedule – call back) or the pomodoro must be abandoned

Tools The creator and his proponents encourage a low-tech approach, using a mechanical timer, paper and pencil. The physical act of winding the timer confirms the user’s determination to start the task; ticking externalises desire to complete the task; ringing announces a break. Flow and focus become associated with these physical stimuli.[1][8]

The technique has inspired application software for several platforms

3.2 Database Structure & Airtable

A database is an organized collection of data.[1] It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Formally, a ”database” refers to a set of related data and the way it is organized. Access to this data is usually provided by a ”database management system” (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized. Because of the close relationship between them, the term ”database” is often used casually to refer to both a database and the DBMS used to manipulate it. Outside the world of professional information technology, the term database is often used to refer to any collection of related data (such as a spreadsheet or a card index). This article is concerned only with databases where the size and usage requirements necessitate use of a database management system.[2]

Airtable is a spreadsheet-database hybrid i.e., the features of a database are applied to a spreadsheet. The fields in an Airtable table are similar to a cell of a spreadsheet, but have types check-boxes, phone numbers, and drop-down lists, and can reference file attachments like images.[1][2] Users can create a database, set up field types, add records, link tables, collaborate with a team, sort the records based on a field and publish views to external websites. When an Airtable database is created, it is automatically hosted to the cloud.[3] The values in the fields are updated real time.

Airtable has six basic components:[12] Bases: All the information needed to create a project is contained in a Base. Bases can be built from existing templates provided by Airtable. In addition, they can also be built from scratch, from a spreadsheet or from an existing Base. Tables: A table is similar to an excel spreadsheet. A Base is a collection of tables.

An example table in a restaurant Base. Views: Views are how we can see a table. Views can be saved for future purposes. Fields: Each entry in a Table is a field. They are not just restricted to hold text. Airtable currently offers 16 basic field types.[13] These are: single-line texts, long text articles, file attachments, check-boxes, single select from drop-down list, multiple-selects from drop-down lists, date and time, phone numbers, email ids, URLs, numbers, currency, percentage, auto-number, formulae and barcodes.[14] Records: Each row of a Table is a Record. Team: Team is a collection of Bases in Airtable. For example, in the adjacent restaurant template which contains all the information we need to store about the restaurants. We can have a 'Restaurants' table to store the names of restaurants along with information about their addresses, ratings, menus, etc. We can have a view to show our favourite restaurants. Each record in the Restaurants table is kept for a particular restaurant. 'Rating' can be kept as a field, to help generate 'My favourite restaurants' view.

3.3 Wiki Pages & Confluence Wiki

A wiki is a website on which users collaboratively modify content and structure directly from the web browser. In a typical wiki, text is written using a simplified markup language and often edited with the help of a rich-text editor.[1] A wiki is run using wiki software, otherwise known as a wiki engine. A wiki engine is a type of content management system, but it differs from most other such systems, including blog software, in that the content is created without any defined owner or leader, and wikis have little implicit structure, allowing structure to emerge according to the needs of the users.[2] There are dozens of different wiki engines in use, both standalone and part of other software, such as bug tracking systems. Some wiki engines are open source, whereas others are proprietary. Some permit control over different functions (levels of access); for example, editing rights may permit changing, adding or removing material. Others may permit access without enforcing access control. Other rules may be imposed to organize content.

dsfdsfds

3.4 V-Model & Agile Methodology

The V-model is a graphical representation of a systems development lifecycle. It is used to produce rigorous development lifecycle models and project management models. The V-model falls into three broad categories, the German Das V-Modell, a general testing model and the US government standard.[2] The V-model summarizes the main steps to be taken in conjunction with the corresponding deliverables within computerized system validation framework, or project life cycle development. It describes the activities to be performed and the results that have to be produced during product development. The left side of the "V" represents the decomposition of requirements, and creation of system specifications. The right side of the V represents integration of parts and their validation.[3][4][5][6][7] However, Requirements need to be validated first against the higher level requirements or user needs. Furthermore, there is also something as validation of system models (e.g. FEM). This can partially be done at the left side also. To claim that validation only occurs at the right side may not be correct. The easiest way is to say that verification is always against the requirements (technical terms) and validation always against the real world or the user needs.

Agile software development describes a set of values and principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams.[1] It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change.[2] The term agile (sometimes written Agile)[3] was popularized by the Agile Manifesto,[4] which defines those values and principles. Agile software development frameworks continue to evolve,[5] two of the most widely used being Scrum and Kanban.[6]

3.5 Version Control with Git & Bitbucket

Git is a version control system (VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development,[8] but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed,[9] data integrity,[10] and support for distributed, non-linear workflows.[11] Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers con-

tributing to its initial development.[12] Its current maintainer since 2005 is Junio Hamano. As with most other distributed version control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server.[13] Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License version 2.

Bitbucket is a web-based hosting service that is owned by Atlassian, used for source code and development projects that use either Mercurial (since launch) or Git (since October 2011[2]) revision control systems. Bitbucket offers both commercial plans and free accounts. It offers free accounts with an unlimited number of private repositories (which can have up to five users in the case of free accounts) as of September 2010. Bitbucket integrates with other Atlassian software like Jira, HipChat, Confluence and Bamboo. It is similar to GitHub, which primarily uses Git. Bitbucket has traditionally tailored itself towards helping professional developers with private proprietary code, especially since being acquired by Atlassian in 2010.[3] In September 2016, Bitbucket announced it had reached 5 million developers and 900,000 teams on its platform.[4] Bitbucket has 3 deployment models: Cloud, Bitbucket Server and Data Center.

sdfdfsdfs

3.6 Microsoft Sharepoint

SharePoint is a web-based, collaborative platform that integrates with Microsoft Office. Launched in 2001,[3] SharePoint is primarily sold as a document management and storage system, but the product is highly configurable and usage varies substantially between organizations. Microsoft states that SharePoint has 190 million users across 200,000 customer organizations.[4]

4 Python & Raspberry Pi

In order to use Raspberry Pi efficiently, I studied Python for a while from a couple of web sites. I mainly focused on Python 3 since it's more up to date than previous version. I tried different codes on Pycharm for Windows before meeting with Linux terminal and Raspberry. Pycharm is one of the most recommended Python IDE's by communities. Here are some of my very

first attempts to use Python.

```
# -----  
# Using Python for the first time!!  
print("Hello Intership!!!")  
  
x = 1  
if x == 1:  
    # indented four spaces, indets Works as brackets in C!  
    print("x is 1.")  
if x==3:  
    print(23)  
    myint = 7 print(myint) # use "#" for commenting  
# -----  
# A sample script that uses lists:  
numbers = [] # creates a list called numbers.  
strings = [] # creates a list called strings.  
names = ["Ali", "Ahmet", "Ayse"] # adds Ali, Ahmet and Ayse to names.  
  
numbers.append(1) # adds "1" to numbers as first element.  
numbers.append(2)  
numbers.append(3)  
  
strings.append("hello")  
strings.append("world")  
  
second_name = names[1]  
print(numbers) # prints [1, 2, 3]  
print(strings) # prints ['hello', 'world']  
print("The second name on the names list is %s" %second_name)  
# prints the second name on the names list is Ahmet!  
# -----  
  
astring = "Hello world!"  
print(astring.index("o")) # prints 4, since o appears firstly at 4th digit.  
print(astring.count("l")) # prints 3, since l appears three times  
print(astring[3:7])  
# prints lo w, starting from 3rd element to 7th element (7th is not included!)
```

```

print(astring[3:7:2])
# prints l, starting from 3rd element to 7th element skipping one character.
print(astring[::-1]) # prints the string reverse.
print(astring.upper()) # prints the string with upper cases.
print(astring.lower()) # prints the string with lower cases.
print(astring.startswith("Hello")) # Returns True
print(astring.endswith("asdfasdf")) # Returns False

# -----
# Using Conditions

if < statement is="" true="" > :
    < do something="" >
    ....
    ....
elif < another statement="" is="" true="" > :
    < do something="" >
    ....
    ....
else:
    < do something="" >
    ....
    ....

# -----
# Using Loops

temurtas = [5, 8, 3, 6]
for halil in temurtas:
    print(halil) # prints every element in temurtas one by one in every loop.
print(temurtas)

count=0
while (count<5) :
    print(count)
    count +=1 else:
    print("count value reached %d" %(count))

```

```

# -----
# Defining Functions

def sum_two_numbers(a, b): # Defining function
    return a + b

x = sum_two_numbers(1,2) # after this line x will hold the value 3!
print("x="
# -----
# Defining Classes

class Vehicle: # define the Vehicle class
    name = ""
    kind = "car"
    color = ""
    value = 100.00

def description(self):
    desc_str = "%s is a %s %s worth $%.2f." % (self.name, self.color, self.kind,
self.value)
    return desc_str

car1 = Vehicle()
car1.name = "Ferrari"
car1.color = "red"
car1.kind = "sport"
car1.value = 600000.00

car2 = Vehicle()
car2.name = "Jeep"
car2.color = "blue"
car2.kind = "SUV"
car2.value = 10000.00

print(car1.description()) # prints Ferrari is a red sport worth $600000.00.
print(car2.description()) # prints Jeep is a blue SUV worth $10000.00.

```

As I went into detail, Python is not very difficult language to learn. In

fact, aside from some indent mistake using it is very simple and clean yet powerfull in various applications.

5 Solar Tracker System Project

dfdsfsfd

6 MATLAB

7 Conclusion

ddsfsf

8 References

dfdsfd