

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ETHERCAT PROTOKOLÜ İLE FB1111-142 KARTI KULLANILARAK
MİKROİŞLEMCİNİN BELİRLENEN KOMUTLARI UYGULAMASI VE
FB1111-142 KARTI İLE HABERLEŞME SAĞLAMASI**

YÜKSEK LİSANS TEZİ

Ömer Faruk ŞAHAN

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

HAZİRAN 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ETHERCAT PROTOKOLÜ İLE FB1111-142 KARTI KULLANILARAK
MİKROİŞLEMCİNİN BELİRLENEN KOMUTLARI UYGULAMASI VE
FB111-142 KARTI İLE HABERLEŞME SAĞLAMASI**

YÜKSEK LİSANS TEZİ

Ömer Faruk ŞAHAN

(518161021)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Ali Fuat ERGENÇ

HAZİRAN 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 518161021 numaralı Yüksek Lisans Öğrencisi Ömer Faruk ŞAHAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “ETHERCAT PROTOKOLÜ İLE FB1111-142 KARTI KULLANILARAK MİKROİŞLEMCİNİN BELİRLENEN KOMUTLARI UYGULAMASI VE FB1111-142 KARTI İLE HABERLEŞME SAĞLAMASI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Dr. Öğr. Üyesi Ali Fuat ERGENÇ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Osman Kaan EROL**
İstanbul Teknik Üniversitesi

Doç. Dr. Özgür Turay KAYMAKÇI
Yıldız Teknik Üniversitesi

Teslim Tarihi : **30 Nisan 2019**

Savunma Tarihi : **12 Haziran 2019**





Geleceğe,



ÖNSÖZ

Bu yüksek lisans çalışmalarını yöneten, yönlendiren, değerli görüş ve eleştirileri ile tez çalışmalarımı destekleyen çok değerli danışman hocam Sn. Dr. Öğr. Üyesi Ali Fuat ERGENÇ 'e teşekkürlerimi bir borç bilirim.

Yüksek lisans tez çalışmasının gerçekleşmesini sağlayan ve bunun için imkân ve olanaklarını sunarak bana destek olan Enerthus Enerji ve Teknoloji A.Ş., Penta Elektronik Telekomünikasyon ve Plastik San. Tic. A.Ş. ve TUSAŞ'a teşekkür ederim.

Nisan 2019

Ömer Faruk ŞAHAN
Mekatronik Mühendisi



İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ	xv
ŞEKİL LİSTESİ	xvii
ÖZET	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Literatür Taraması	2
2. KULLANILAN DONANIMLAR ve YAZILIMLAR	5
2.1 Bilgisayar	5
2.2 TwinCAT Yazılımı	5
2.2.1 TwinCAT gerçek zaman ayarları	6
2.2.2 FB1111-142 kartının TwinCAT'e eklenmesi	8
2.3 Arduino IDE	12
2.4 FB1111-142 Piggyback	12
2.4.1 Gösterge LEDleri	13
2.4.2 Veri işleme arayüzü (PDI)	13
2.4.3 Güç kaynağı ve bağlantılar	13
2.4.4 Komponent yerleşim görüntüsü	14
2.4.5 Güç test padleri	15
2.5 Arduino Due	15
3. YAZILIM	17
3.1 Genel Yapı	17
3.1.1 Veri Yapısı	17
3.2 Başlama Algoritması	18
3.3 Döngüsel Çalışma	20

3.4 Arayüz	22
4. SONUÇ VE ÖNERİLER.....	25
KAYNAKLAR.....	27
EKLER	29
ÖZGEÇMİŞ.....	31



KISALTMALAR

PLC	: Programmable Logic Controller
PWM	: Pulse Width Modulation
PTO	: Pulse Train Output
EtherCAT	: Ethernet for Control Automation Technology
ESC	: EtherCAT Slave Controller
PDI	: Process Data Interface
IDE	: Integrated Development Environment
ADC	: Analog Digital Converter
MAC	: Media Access Controller
CAN	: Controller Area Network
TDK	: Türk Dil Kurumu
Mbps	: Megabits per second
MS-DOS	: Microsoft Disc Operating System
CPU	: Central Process Unit
GPIO	: General Purpose Input Output
PIO	: Parallel Input Output Controller
KHz	: Kilo Hertz



SEMBOLLER

us : Mikro Saniye





ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Test pinleri.....	15
Çizelge 3.1 : Komut bitleri ve anlamları.	18





ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Ağ üzerinde veri çerçevesinin dolaşımı.....	2
Şekil 1.2 : EtherCAT veri yapısı.....	3
Şekil 2.1 : Öncelik ataması.	6
Şekil 2.2 : Gerçek zaman sekmesi [14].....	6
Şekil 2.3 : Gerçek zamanlı çalışma için seçilmiş çekirdekler [14].	7
Şekil 2.4 : CPU limit ayarları [14].	7
Şekil 2.5 : Latency(gecikme) ayarı [14].....	8
Şekil 2.6 : Yeni öge ekleme.	8
Şekil 2.7 : IO aygıtını eklemek için tarama yapılması.	9
Şekil 2.8 : Tarama butonu.	9
Şekil 2.9 : Bulunan IO aygıtlarının listesi.....	9
Şekil 2.10 : Donanımlar için tarama seçeneği.	9
Şekil 2.11 : Donanımın eklenmesi.	10
Şekil 2.12 : FB1111-142 kartı portları.	11
Şekil 2.13 : Çevrimiçi olarak değer okuma ve yazma.	11
Şekil 2.14 : FB1111-14X kartı [16].	12
Şekil 2.15 : Pin numaraları ve güç girişleri.....	14
Şekil 2.16 : Üstten görünüş.	14
Şekil 2.17 : Altan görünüş.	15
Şekil 2.18 : Arduino Due.	16
Şekil 3.1 : Genel haberleşme yapısı.	17
Şekil 3.2 : TwinCAT başlama akış şeması.	19
Şekil 3.3 : Arduino Due başlama akış şeması.	20
Şekil 3.4 : OUT_VALID zamanlaması [17].	20
Şekil 3.5 : Arduino Due durum makinesi.....	21
Şekil 3.6 : TwinCAT durum makinesi.	22
Şekil 3.7 : TwinCAT arayüzü.	23
Şekil 3.8 : Frekans hesaplama algoritması.	24
Şekil 4.1: İstenmeyen hata.	25

Şekil 4.2 : Merkez hizalı PWM - PTO sinyali kesmesi [17].	26
Şekil 4.3 : Sola hizalama ve polaritenin değiştirilmesi [17].	26
Şekil 4.4: 500KHz’de 5 adet puls verilmesi.	26



ETHERCAT PROTOKOLÜ İLE FB1111-142 KARTI KULLANILARAK MİKROİŞLMECİNİN BELİRLLENEN KOMUTLARI UYGULAMASI VE FB1111-142 KARTI İLE HABERLEŞME SAĞLAMASI

ÖZET

EtherCAT Beckhoff Otomasyon tarafından geliştirilen gerçek zamanlı endüstriyel ethernet teknolojisidir. EtherCAT protokolünde master telegram gönderir. Gönderilen telegram bütün düğümlerden geçer. Her düğüm, telegramı okurken kendi adresine gerekli dataları ‘on the fly’ olarak yazar. Böylece haberleşme sürecindeki gecikme sadece donanım üzerindeki propagasyon kadar olur. Son düğümdeki donanım, çıkış portunun takılı olmadığını anlayarak, telegramı tekrar mastera ethernet teknolojisinin full-duplex özelliğini kullanarak gönderir. Bu sayede teorik olarak 100Mbit/s’den hızlı olarak data aktarımı gerçekleştirilebilir.

Bu tez kapsamında bir kara aracının motoru üzerindeki belirli bir ihtiyacı karşılayacak şekilde tasarım yapılmıştır. Oluşturulan sistemde master olarak bir ana bilgisayar, ana bilgisayar ile EtherCAT protokolüyle haberleşen slave FB1111-142 kartı, FB1111-142 kartı ile paralel olarak haberleşen ve bu paralel haberleşmede slave olarak görev yapan Arduino Due kartı bulunmaktadır. Ana bilgisayar tarafından kontrol edilen bir alt sistemin, ana bilgisayar ile eş zamanlı olarak haberleşmesi, verilen komutları işlemesi, yapılan ölçüm sonuçlarını aktarması beklenmektedir.

TwinCAT programı ile 4 çekirdekli bir bilgisayarın 2 çekirdeğinin atamasının yapılmasıyla birlikte sıradan bir bilgisayar, PLC gibi davranarak gerçek zamanlı olarak çalışmaktadır. Böylece ana bilgisayar olarak kullanılabilir.

Ana bilgisayar EtherCAT protokolü üzerinden gerçek zamanlı olarak FB1111-142 piggyback kartı ile haberleşmektedir. FB1111-142 kartı gelen dataları dijital çıkışlar üzerinden paralel olarak Arduino Due kartına aktarmaktadır. Aynı zamanda dijital girişler ile Arduino Due kartından paralel olarak ilgili dataları, verilmiş komutların sonuçlarını almaktadır. FB1111-142 kartı Arduino Due kartı ile ana bilgisayar arasında köprü kurarak işlemlerin yapılmasını sağlamaktadır.

Arduino Due kartı gelen datalara göre ADC ölçümü, PWM ve PTO üretimi yapmaktadır. Ayrıca gelen komutlara göre IO çıkışı veya girişini de işlemektedir.

Haberleşme gereksinimi olarak FB1111-142 üzerindeki 16 bit çıkış - 16 bit giriş olarak kullanılmaktadır. Bu çıkışlardan; Arduino Due’ye adres, komut ve data verilmektedir. Aynı şekilde girişlerden de sonuçlar alınmaktadır. Her 16 bitin 5 biti adres biti, 3 biti komut biti, 8 biti veri olarak tanımlanmıştır. 8 bitten daha fazla veri için komut biti yardımıyla 2 High Byte – Low Byte olarak olarak 2 parça halinde gönderime imkan tanınmıştır.

Sonuç olarak oluşturulan veri haritası ile haberleşme gerçekleştirilmiştir. Verilen komutlara uygun olarak PWM, PTO çıktısı sağlanmıştır. ADC ölçümleri

gerçekleştirilerek veriler gönderilmiştir. Kablo kopukluğuna önlem olarak geliştirilen başlama algoritmasıyla tüm kabloların bağlı olduğu kontrol edilerek teyit işlemi yaptırılmıştır.



APPLICATION OF THE MICROPROCESSOR'S DETERMINED COMMANDS USING THE ETHERCAT PROTOCOL BY FB1111-142 CARD AND COMMUNICATION WITH THE FB111-142 CARD

SUMMARY

The system is a collection of items or components that are organized for a common purpose. The system has a function which includes inputs and outputs. The difference between the system and the real-time system is that the real-time system guarantees the outputs will be ready within the specified time constraint. There are three real-time system types, hard real-time, firm real-time and soft real-time. Hard real-time systems cannot tolerate to miss a deadline. In hard real-time systems, missing a deadline results in catastrophic consequences. In firm real-time systems, missing a deadline may not cause catastrophic consequences but could cause undesired consequences. For example, reduction of huge quality of product. In soft real-time systems, missing deadlines are tolerable but not desired. The most important components of real-time systems are time constraint, latency, priority and being deterministic.

EtherCAT is a real-time industrial ethernet technology developed by Beckhoff Automation. The master sends telegram to slaves in the EtherCAT protocol. The telegram sent passes through all slave nodes. Each node reads the telegram and writes the necessary data to its address 'on the fly'. Thus, the delay in the communication process is only as much as propagation on the hardware. The hardware on the last node recognizes that the output port is not plugged in, then sends the telegram again to master, using the full-duplex feature of the ethernet technology. In this way, theoretically, data transfer can be performed faster than 100Mbit/s.

Within the scope of this thesis, a design has been designed to meet a specific need on the engine of a land vehicle. In the generated system, there is a master computer, the slave FB1111-142 board communicating with the host computer and the EtherCAT protocol, and the Arduino Due board, which operates in parallel communication with the FB1111-142 board. It is expected that a subsystem controlled by the host will simultaneously communicate with the host, process the given commands, and transmit the measured results.

In applications with spatially distributed processes requiring simultaneous actions, exact synchronization is particularly important. The ethercat communication is based on distributed clocks for synchronization of nodes. At the hardware level, the clocks on the nodes are adjusted according to the reference clock. Calibration is performed according to the reference in each cycle and the result is lower than 1us to jitter level. The signal sent from the reference clock is delayed to reach the slaves. This delay time can be measured during the network start-up period or by continuously measuring the clocks under 1us relative to each other.

The FB1111-142 board is equipped with an ET1100 processor. The processor processes the incoming EtherCAT data and sets the outputs according to the incoming data. At the same time, the input port values are sent to the host by typing in the

telegram. Updating the outputs takes place with four ways. First way is at end of the EtherCAT frame which is used in thesis. Second way is 'Distributed Clocks SYNC 0' events. 'Distributed Clocks SYNC 1' and last way is at the end of EtherCAT frame which triggered by process data watchdog. In addition, to observe the outputs data from pins, the following condition should be provided; logic 1 must be applied to the output enable pin, SyncManager watchdog must be either active or disabled, output values have to be written register within a valid EtherCAT frame and configured output update event must be occurred for the outputs to be active on the FB1111-142 board. An output event is always signaled by a pulse on 'OUT_VALID' even if the digital outputs are not changed.

With the assignment of 2 cores of a 4-core computer with TwinCAT program; an ordinary computer works in real time by acting as a PLC. If the assigned kernels are shared with Windows, the kernels serve TwinCAT within the set CPU limits. During the period outside the CPU limits, the CPUs are running on the Windows operating system. Since the TwinCAT program performs CPU settings over the kernel, there is no link between the operating system and the CPU run times. That is, the operating system cannot assign a task to the kernels. In real time systems, the specified cycle times are completed with the scheduler and real time requirements are provided. Real-time systems programmed on TwinCAT usually end in much less time than the specified time intervals. Thanks to the intelligent programming of TwinCAT, the time remaining from the end of the job is transferred to Windows and system performance is improved. In addition, warnings can be given by setting latency times via TwinCAT. With all these features, an ordinary computer can be used as a host that provides real-time features.

The host communicates with the FB1111-142 piggyback board in real time via the EtherCAT protocol. The FB1111-142 board transmits incoming data in parallel to the Arduino Due board via digital outputs. At the same time, the corresponding inputs from the digital inputs and Arduino Due board are the results of the given commands. The FB1111-142 board is a bridge between the Arduino Due board and the host computer, allowing transactions to be made. The Arduino Due board produces ADC measurement, PWM and PTO according to the data. It also processes IO output or input according to incoming commands.

On Arduino Due Atmel 32 bit SAM3X8E ARM Cortex-M3 processor. There are 54 digital inputs and outputs and 12 analog inputs. 12 of the digital input-outputs can be used as PWM output. It has a clock speed of 84 MHz and features up to 512 Kbytes of flash and up to 100Kbytes SRAM. The peripheral hardware consists of a High Speed USB Host and Device port with embedded transceiver, an Ethernet MAC, 2 CANs, a High Speed MCI for SDIO/SD/MMC, an External Bus Interface with NAND Flash Controller (NFC), 5 UARTs, 2 TWIs, 4 SPIs, a PWM timer, three 3-channel general-purpose 32-bit timers, a low-power RTC, a lowpower RTT, 256-bit General Purpose Backup Registers, a 12-bit ADC and a 12-bit DAC. In addition, Up to 103 input - output lines with external interrupt capability (edge or level sensitivity), debouncing, glitch filtering and ondie Series Resistor Termination. The FB1111-142 board gives the pulse to the 'OUT_VALID' pin when it sets its outputs according to the incoming data. The rising edge of the 'OUT_VALID' pulse enters the Arduino Due board port interrupt. It checks that the results of the previous command are given to the outputs in the port interrupt. Then the input ports are read and the data is received.

Atmel SAM3X has two types of input output lines. These are general purpose inputs and outputs and system inputs and outputs. General purpose input and outputs can have alternate functions with the multiplexing capabilities of the Parallel Input Output controller. The same PIO line can also be used in input output mode or by a multiplexed peripheral. System inputs outputs includes pins, such as test pins, oscillators, erasing or analog inputs. General purpose input output lines are controlled by parallel input output controller. All inputs outputs have several input or output modes such as pull-up, input Schmitt triggers, multi-drive (open-drain), glitch filters, debouncing or input change interrupt. Programming of these modes is performed independently for each inputs outputs line through the PIO controller user interface. Each input output line is dedicated to a bit number from 32 bit registers of 32 bit user interface.

Parallel input output controller attributes are; An input change interrupt enabling level change detection on any input output line. Additional Interrupt features enabling rising edge, falling edge, low level or high level detection on input output line. A glitch filter providing rejection of glitches lower than one-half of system clock cycle. A debouncing filter providing rejection of unwanted pulses from key or push button operations. Multi-drive capability similar to an open drain input output line. Control of the pull-up of the input output line. Input visibility and output control. The parallel input output controller also enables a synchronous output providing up to 32 bits of data output in a single write operation. Especially synchronous output feature is a key function for parallel communication by using general purpose input outputs. With this function desired cycle times can be reached.

To using input and outputs with parallel input output controller, PIO_PER (PIO Enable Register) and PIO_PDR (PIO Disable Register) registers should be written. Result could be controlled by reading PIO_PSR (PIO Status Register) register. A value of set indicates the pin is controlled by PIO Controller.

To reading inputs level from PIO controller requires to read PIO_PDSR (Pin Data Status Register). The register provides the values regardless of whether the pins configured as inputs or outputs. The result of configuration could be checked from PIO_OSR (Output Status Register).

The levels of outputs can be set by writing PIO_SODR (Set Output Data Register) and PIO_CODR (Clear Output Data Register).

For synchronous data output, firstly PIO controller configuration should be made. By setting PIO_OER (Output Enable Register) and PIO_ODR (Output Disable Register) registers the pins are configured as driven by PIO controller. Setting output by PIO_SODR and PIO_CODR requires two successive write operations into two different register. To overcome this, the PIO Controller provides a direct control of PIO outputs by single write access to PIO_ODSR (Output Data Status Register). Only bits unmasked by PIO_OWSR (Output Write Status Register) are written. The mask bits in PIO_OWSR are set by writing to PIO_OWER (Output Write Enable Register) and cleared by writing to PIO_OWDR (Output Write Disable Register).

The 16-bit output on FB1111-142 is used as a 16-bit input as a communication requirement. From these outputs; Arduino Due is provided with address, command and data. Similarly, the results are taken from the inputs. 5 bits of each 16 bits are defined as address bits, 3 bits are command bits and 8 bits are defined as data. For more than 8 bits, 2 High Bytes - Low Bytes can be sent with 2 bits.

According to the established communication protocol, the parameters are read in 5 bit address, the commands are executed with 3 bit command and in the 8 bit data part according to the command or address. 8 different commands can be sent with a 3-bit command. The commands are as follows; no operation, read analog digital converter, set pulse time output, logic input value read, write logic output value.

As a result, communication with the generated data map has been carried out. PWM, PTO output is provided according to given commands. Data were sent by performing ADC measurements. With the start-up algorithm developed as a precaution to the cable disconnection, all the cables are checked and confirmed.



1. GİRİŞ

Günümüzde kullanılan motorlu taşıtların bir çoğunda Elektronik Kontrol Ünitesi (ECU) bulunmaktadır. Elektronik Kontrol Üniteleri, araç üzerindeki aktüatörleri, ateşleme zamanlarını, sensör verilerini kontrol ederek; aracın üst düzey performansta verimli ve güvenli olarak çalışmasını sağlar. Araç üzerinde manifold mutlak basınç sensörü, klape pozisyon sensörü, eksoz gaz dönüşüm sensörü, soğutucu sıcaklık sensörü, debi sensörü, oksijen sensörü gibi sensörler bulunmaktadır. Bu sensörlerden bir çoğundan analog ölçüm almaktadır [1].

Bir otomobil içinde yaklaşık 3860 metre bakır kablo, 45 adeti ağa bağlı olmak üzere 61 ECU, 11.136 elektriksel parça ve 3 ayrı ağ hattı bulunmaktadır [2]. Araç üzerinde bulunan sistemlerin kullandığı sensörlerden doğru veri alması önem arz etmektedir. Bu nedenle analog bir sinyalin aracın içerisindeki gürültü altında bir noktadan bir noktaya iletmek makul bir çözüm değildir. Ölçülen analog verinin kısa mesafe içerisinde dijital veriye dönüştürülmesi gerekmektedir.

1.1 Tezin Amacı

Bu tez çalışmasında bir araç elektronik kontrol ünitesi simule edilerek mikrokontrolör ile haberleşme sağlanarak, ADC ölçümü, PWM çıkışı, PTO çıkışı sağlanmıştır.

Araç elektronik kontrol ünitesinin simülasyonu için bilgisayar üzerinde TwinCAT isimli program kurulmuştur. Bu program sayesinde 2 adet çekirdek Windows işletim sisteminden izole edilerek TwinCAT programına adanmıştır. Böylece yazılan PLC kodu gerçek zamanlı olarak elektronik kontrol ünitesinin simülasyonuna imkan tanımıştır.

Bilgisayar ile FB1111-142 piggyback kartı ethernet üzerinden EtherCAT protokolünü kullanarak haberleşmektedir.

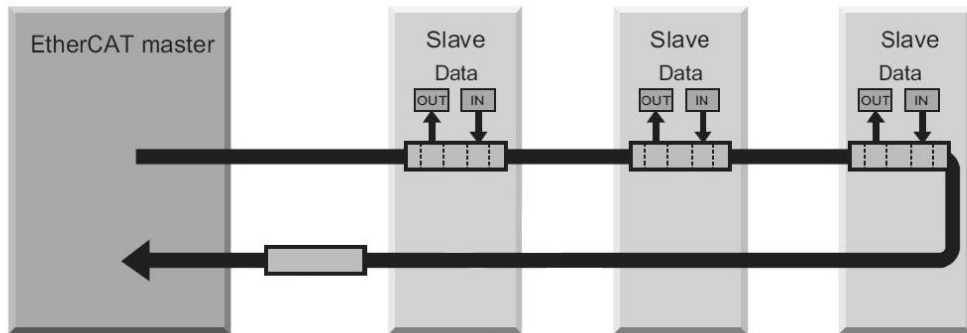
FB1111-142 kartı üzerinde 32 adet GPIO bulunmaktadır. Bu GPIO'ların 16 tanesi giriş 16 tanesi çıkış olarak ayarlanmıştır. Her 16 GPIO için 5 adet adres, 3 adet komut

ve 8 adet veri olarak tanımlanmıştır. Bu sayede FB1111-142 kartı ile Arduino paralel haberleşme ile karşılıklı olarak veri transferini gerçekleştirmektedir. Gelen komutlara ve adreslere göre sistemin tanımlanmış işlemleri yerine getirerek görevini ifa etmesi beklenmektedir.

1.2 Literatür Taraması

EtherCAT; Beckhoff Otomasyon'un geliştirdiği, Ethernet teknolojisi üzerine kurulmuş bir haberleşme protokolüdür. EtherCAT protokolü IEC 61158 standartlarına uygundur [3]. Otomotiv sanayinin ihtiyacı olan 'hard' ve 'soft' gerçek zaman isterlerini karşılayabilmektedir [1].

EtherCAT master ve slave ilişkisi ile çalışmaktadır. Masterın gönderdiği mesajlar bütün slavelerden geçerek dolaşır. Bu dolaşım esnasında her slave kendi adresine mesaj kesintiye uğratılmadan kendi verilerini okur ve yazar. Mesajdaki tek gecikme elektronik komponentler üzerindeki yayılma gecikmesidir. Son slave kendisinden sonra slave olmadığını anlayarak mesajı tekrar hatta yazar (Şekil 1.1). Böylece full-duplex haberleşme sağlanmış olur. EtherCAT teorik olarak bant genişliğinin verimini %90'ın üzerine çıkarabilmektedir. Böylece 100 Mbps üzerinde haberleşme yapılabilir. Aynı anda hem okuma hem de yazma yapılması durumunda bu hız 2 katına çıkmaktadır.

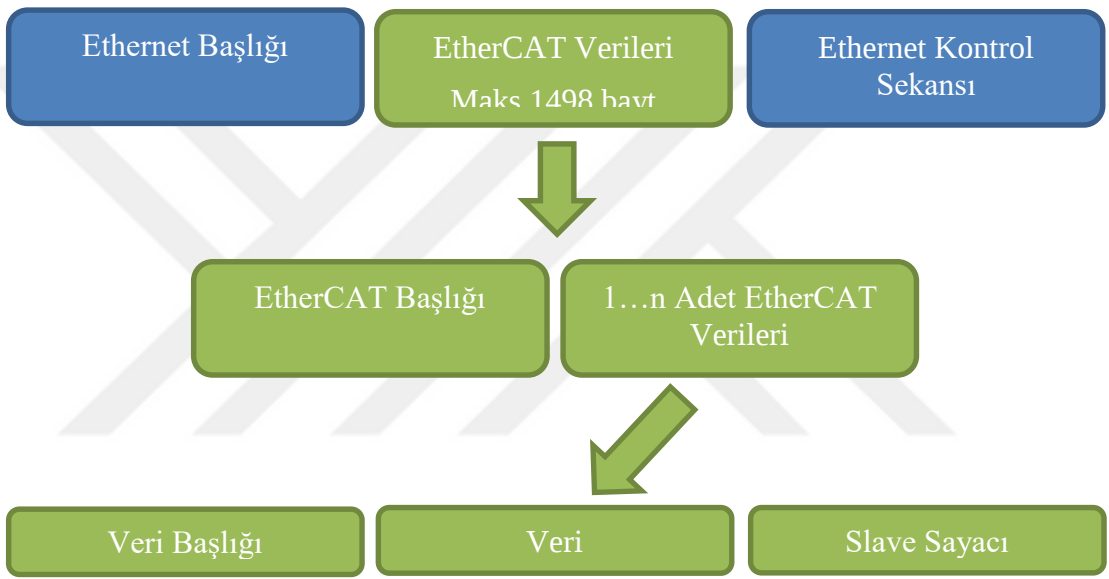


Şekil 1.1 : Ağ üzerinde veri çerçevesinin dolaşımı.

EtherCAT haberleşmesinde sadece master yeni bir haberleşme çerçevesi gönderebilmektedir. Slavelerin hepsi sadece gelen çerçeveyi iletmektedir. Bu konsept sayesinde beklenmedik gecikmeler önlenmiş ve gerçek zaman yeteneği garantilenmiştir.

EtherCAT teknolojisinde master ekstra haberleşme işlemcisi kullanmadan standart Ethernet MAC kullanmıştır. Böylece ethernet portu olan donanım ile işletim sisteminden ve yazılımından bağımsız olarak haberleşebilmektedir. EtherCAT slave aygıtları EtherCAT Slave Controller (ESC) kullanarak, donanım üzerinde verileri işler ve yazar. Böylece ağ performansı slave aygıtlarından bağımsız ve deterministik olmuştur.

EtherCAT haberleşmesi Ethernet protokolü üzerine kurulmuştur. Bu yüzden haberleşme verilerindeki yapı Ethernet başlığı, EtherCAT başlığı, EtherCAT telegramı, Ethernet kontrol baytları şeklindedir (Şekil 1.2).



Şekil 1.2 : EtherCAT veri yapısı.

EtherCAT haberleşmesi yüksek çözünürlüklü senkronizasyon için dağıtılmış saatleri kullanır. Master bütün slaveler için referans saat oluşturur. Slaveler de donanım seviyesinde ölçümlerle kendi saatlerini düzenlerler. Bu mekanizmayla jitter süresi 1 us altında kalmaktadır. Yapılan çalışmada; dağıtılmış saat mekanizmasının, ağ çalışmaya başladıktan bir kaç saniye sonunda senkronizasyonun çok iyi bir şekilde sağlandığı kanıtlanmıştır [4].

Ethernet tabanlı endüstriyel olarak kullanılan protokoller: EtherCAT, Powerlink, Ethernet/IP, Modbus/TCP ve ProfiNet'tir. Literatüre bakıldığında toplamda EtherCAT'in avantajlı olduğu görülmektedir [5]. ProfiNet'in asenkron haberleşmede EtherCAT'e göre üstün olduğu görülmektedir [6]. Ethernet/IP ve Modbus/TCP deterministik değildir ve bu yüzden 'hard' gerçek zaman için uygun değildir.

EtherCAT'in ProfiNet ve Powerlink karşısındaki en büyük avantajı uygulama yapmanın maliyeti ve teknolojisinin yaygınlığıdır [7].

CAN maliyeti düşük gömülü ağ sistemleri için uygunluğunu korumaktadır. Ethernet tabanlı ağ yapılarının artık CAN'ın yetersiz kaldığı (sınırlı baudrate ve ağ uzunluğu) noktaların üstesinden gelebilmektedir. Ancak CAN uygulama maliyetinin düşüklüğü ve düşük kaynak gereksinimi sayesinde otomotiv, havacılık, küçük gömülü sistem uygulamalarında halen tercih edilmektedir. EtherCAT'in CAN'e göre avantajları: veri aktarma miktarı, sınırsız ağ uzunluğu, artırılmış sistem performansı, donanım seviyesinde çözümlerdir [8].

Yapılan bir başka çalışmada gerçek zamanlı olarak 6 eksenli bir robotun EtherCAT protokolü kullanılarak control edilmesi sağlanmıştır. Bu çalışmada master kontrollör olarak slaveler ise veri işleyici olarak kullanılmıştır. Sistemin 'on the fly' data işlemesinden ve 100 Mbps ile haberleşmesinden dolayı çok kısa tepki süresi olduğu değerlendirilmiştir. Çalışmada 10Khz data transfer ve 1Khz pozisyon kontrol frekansında, robot kontrol sisteminin çalışabildiğini ispatlamıştır [9].

EtherCAT teknolojisi endüstride yaygın bir yer kaplamaktadır. Farklı bir alan olarak rüzgar türbinlerinde kullanılmıştır. Yapılan bir çalışmada, EtherCAT tabanlı olarak rüzgar türbininin ana kontrol sisteminin tasarımı anlatılmıştır. Bilgisayar kontrol teknolojisi ve ağ kontrolü rüzgar türbinlerini güvenilirliğini ve verimli çalışmasını sağlamaktadır [10].

2. KULLANILAN DONANIMLAR VE YAZILIMLAR

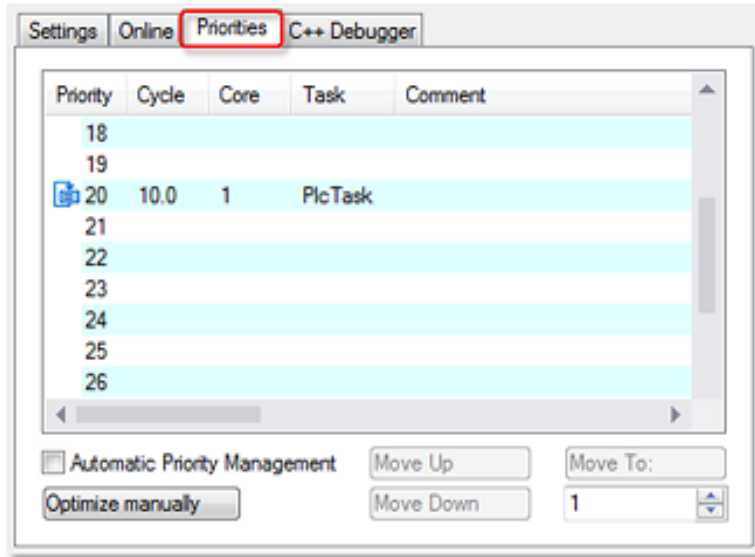
2.1 Bilgisayar

TDK tanımına göre; çok sayıda aritmetiksel veya mantıksal işlemlerden oluşan bir işi, önceden verilmiş bir programa göre yapıp sonuçlandıran elektronik araç, elektronik beyin olarak adlandırılan bilgisayarlar günümüzde hayatın vazgeçilmez bir parçası haline gelmiştir. Bilgisayarların tarihsel gelişimine bakıldığında, ilk başlarda bir oda büyüklüğünde iken günümüzde kol saati boyutlarına kadar küçülmüştür. Bilgisayarı hesap makinesinden ayıran en büyük özelliklerden birisi de istenilen yazılımın saklanarak istenilen zamanda tekrar çalıştırılabilmesidir [11].

Kullanılan bilgisayarın özellikleri: intel i7-7700HQ 2.8 Ghz, 8 GB RAM

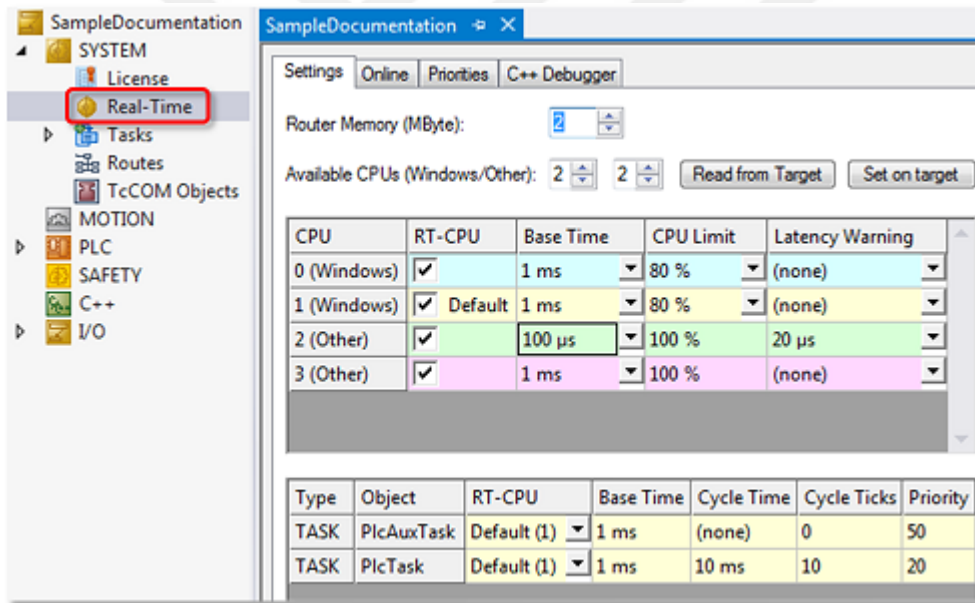
2.2 TwinCAT Yazılımı

Gerçek zamanlı sistemler verilen bir hesaplama işini belirli bir süre içerisinde gerçekleştirerek doğru sonuçları veren deterministik sistemlerdir [12] [13]. Bilgisayar üzerindeki gerçek zamanlı fonksiyonlar temel olarak zamanlayıcı kesmesini baz alır. Bu zamana bağlı olan kesme, bilgisayar mimarisinin gelişimin geçmişine bağlıdır ki MS-DOS ile başlayan gerçek zamanlı kesme günümüz bilgisayarlarında halen aktif durumdadır [13]. Windows NT'nin endüstriyel alanda tutulmasıyla ve kararlılığıyla birlikte TwinCAT programının temellerini oluşturmuştur. Windows bir çok işlemi bir paralel olarak yürütmektedir. Bu yüzden Windows üzerine gerçek zamanlı bir iş yaptırmak için kernel üzerinden zamanlı kesme ayarlanmıştır. Bu yöntem ile Beckhoff 'double-tick' isimli patenti almıştır. Ayrıca TwinCAT ile gerçek zamanın sac ayaklarından biri olan öncelik ve gecikme atamaları da yapılabilmektedir (Şekil 2.1).



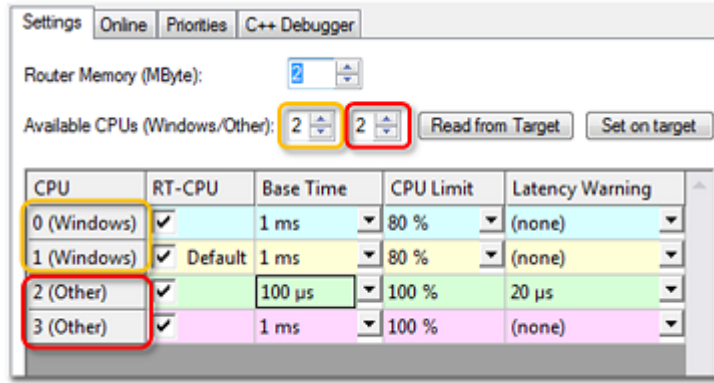
Şekil 2.1 : Öncelik ataması.

2.2.1 TwinCAT gerçek zaman ayarları



Şekil 2.2 : Gerçek zaman sekmesi [14].

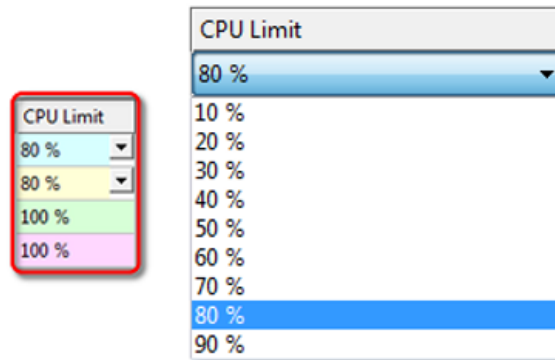
Şekil 2.2’de gösterildiği gibi TwinCAT’de proje oluşturduğumuzda gerçek zaman sekmesinden çekirdek ayarları yapılabilmektedir. 4 çekirdek bir bilgisayar için 2 tanesini TwinCAT için atarken 2 tanesini Windows’a bırakılmaktadır (Şekil 2.3).



Şekil 2.3 : Gerçek zamanlı çalışma için seçilmiş çekirdekler [14].

TwinCAT kernel modunda zamanlayıcı ile çalıştığı için CPU zamanını Windows ile kendisi arasında paylaşır. ‘Base Time’ sürekli olarak kesmenin geleceği zamanı belirtir. Bu zaman aralıklarında Windows duraklatılır ve TwinCAT’ın gerçek zaman taskları için zaman ayrılır¹.

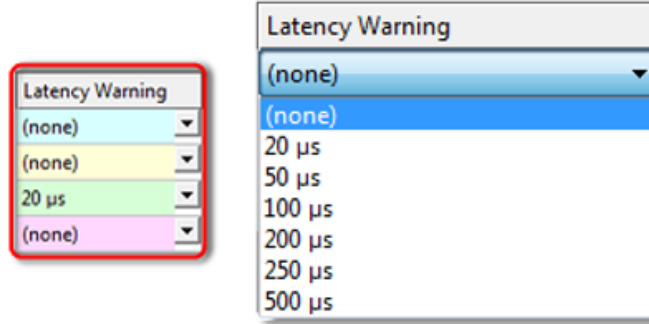
CPU limit; zamanlayıcı tarafından ayrılan ‘Base Time’ süresinin, yüzde kaçının TwinCAT tarafından kullanılabileceğini ayarlar. Bu aşamada gerçek zaman çevrimi bittiği anda zamanlayıcı tarafından kalan süre Windows’a verilir. Böylece kalan süre bloklanmadan verimli bir şekilde kullanılmış olur. CPU limitin yüksek ayarlanması tavsiye edilmektedir (Şekil 2.4). Gerçek zamanlı işlemler, genellikle kendilerine ayrılan sürenin küçük bir kısmında işlemlerini bitirmektedir [14].



Şekil 2.4 : CPU limit ayarları [14].

¹ 1 ms altındaki süreler Windows’un performansını çok düşüreceği için tavsiye edilmemektedir.

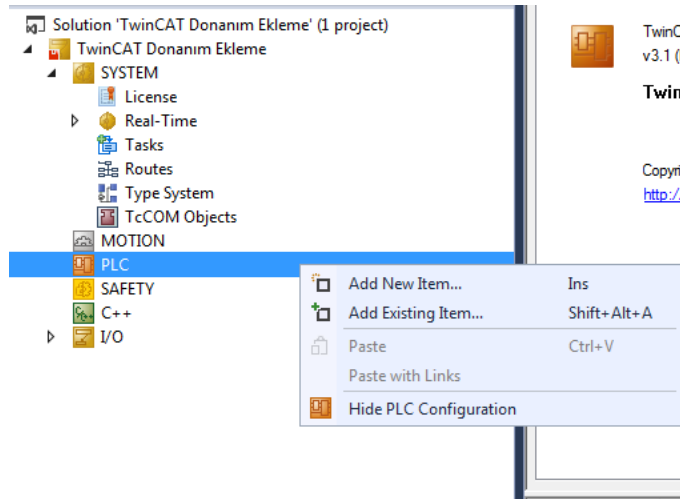
TwinCAT gerçek zaman düşük seviye gecikmeyle çalışmaktadır. Ancak donanıma bağlı olarak gecikmeler oluşabilmektedir. TwinCAT ayarların yapımı durumunda sistemde oluşacak bir gecikmeyi kendisi belirleyerek uyarı verebilmektedir (Şekil 2.5).



Şekil 2.5 : Latency(gecikme) ayarı [14].

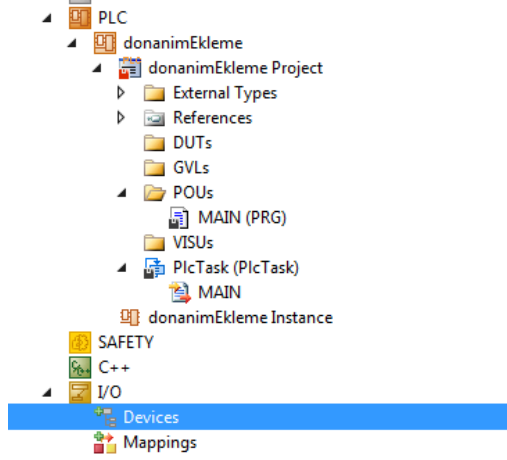
2.2.2 FB1111-142 kartının TwinCAT'e eklenmesi

Öncelikle TwinCAT programı başlatılmalıdır. Visual Studio arayüzüne entegre edilmiş TwinCAT programı ile proje açılmalıdır. CPU atamaları yapıldıktan sonra PLC eçeneği üzerine sağ tıklanarak yeni öge eklenmelidir (Şekil 2.6). Bu aşama 'Standart PLC Project' seçeneği seçildikten sonra proje oluşturma işlemi tamamlanmış olacaktır. Program 'POU' klasörü altındaki 'MAIN' dosyası üzerinden yazılmaktadır.



Şekil 2.6 : Yeni öge ekleme.

Proje oluşturulduktan sonra 'IO' seçeneği altından 'Devices' seçiminin ardından donanımlar için tarama 'Scan' ikonuna tıklanmalıdır (Şekil 2.7) (Şekil 2.8).

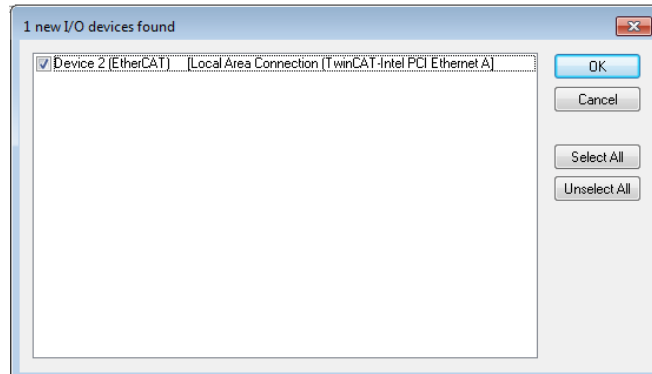


Şekil 2.7 : IO aygıtını eklemek için tarama yapılması.

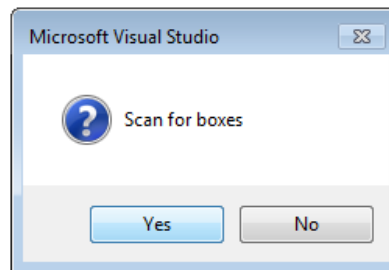


Şekil 2.8 : Tarama butonu.

Tarama yapıldıktan sonra Şekil 2.9'da gösterildiği gibi aygıtların liste halinde görülmesi gerekmektedir. Uygulamada EtherCAT aygıtının bağlı olduğu görülmektedir. Ardından donanımlar için tarama seçeneği gelmektedir (Şekil 2.10).

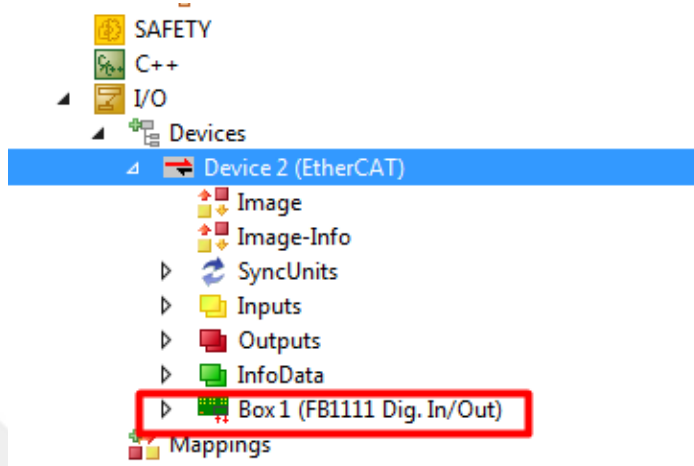


Şekil 2.9 : Bulunan IO aygıtlarının listesi.



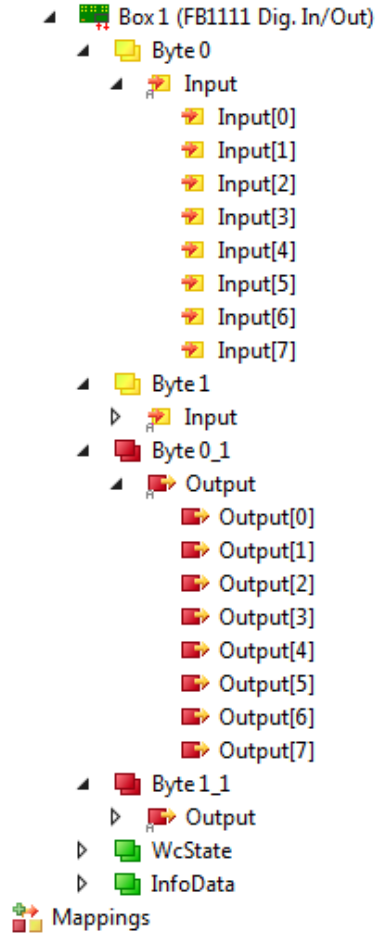
Şekil 2.10 : Donanımlar için tarama seçeneği.

Donanımlar için tarama işleminin sonunda, başarılı şekilde eklenen donanım Şekil 2.11’de kırmızı kutu içerisinde gösterildiği gibi listede görünmektedir. Eklenen donanım portları Şekil 2.12’deki gibi görüntülenebilmektedir. Üzerine çift tıklayarak port atamaları ve çevrimiçi olarak değerleri görülüp yazılabilmektedir² (Şekil 2.13).

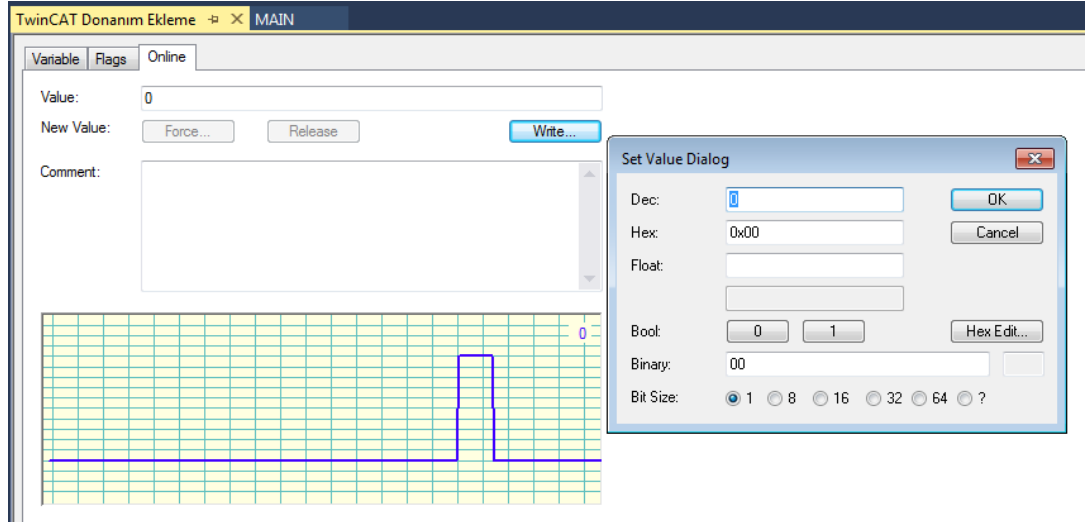


Şekil 2.11 : Donanımın eklenmesi.

² FB1111-142 kartı OE_EXT (Output Enable) pinine lojik 1 uygulanmadığı sürece, çıkış portları yüksek empedans gösterecektir.



Şekil 2.12 : FB1111-142 kartı portları.



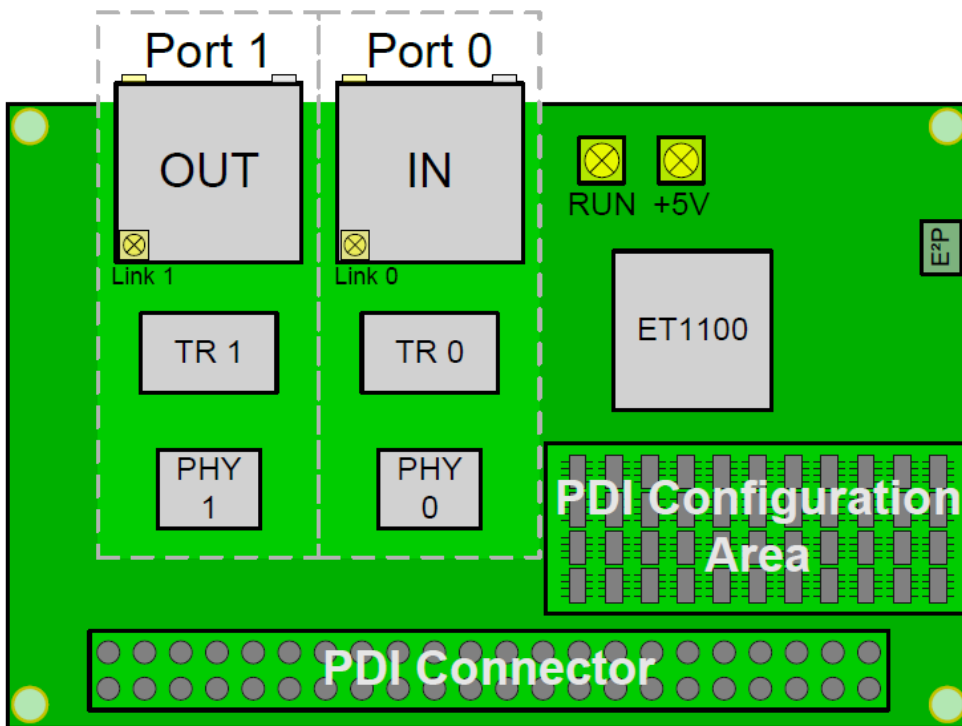
Şekil 2.13 : Çevrimiçi olarak değer okuma ve yazma.

2.3 Arduino IDE

Arduino IDE java programında geliştirilmiş, cross-platform (Windows, macOS, Linux) bir programlama arayüzüdür. Bu IDE ‘Processing’ ve ‘Wiring’ adlı iki geliştirme ortamından oluşturulmuştur. IDE içinde kopyalama, yapıştırma, otomatik girinti ayarlama, sintaks renklendirme, arama ve değiştirme, parantez eşleme gibi özellikler bulunmaktadır. Açık kaynak kodludur. Arduino IDE C ve C++ dillerini özel kod yapısı kullanarak desteklemektedir. ‘Wiring’ üzerinden IDE setup ve loop isimli iki temel fonksiyondan oluşan yazılım kütüphanesi sağlamaktadır [15].

2.4 FB1111-142 Piggyback

EtherCAT Piggyback kartı FB1111-14X; ET1100 slave işlemcisi , 2 ethernet portu ve PDI konfigürasyon alanından oluşan devre kartıdır (Şekil 2.14). Mikroişlemci, SPI ve IO arayüzünü sağlamak için 3 çeşit PDI konfigürasyon şekli bulunmaktadır.



Şekil 2.14 : FB1111-14X kartı [16].

Devre kartı yukarıda görüldüğü üzere port 0, port 1, PDI konfigürasyon alanı, EEPROM, LEDler, PDI konnektör alanı ve ET1100 slave işlemcisi olarak yapısı oluşturulmuştur [16].

2.4.1 Gösterge LEDleri

FB1111 kartı üzerinde 4 adet gösterge LEDi bulunmaktadır. Bunlar;

- +5V : 5 volt güç göstergesi
- RUN : Uygulama durum göstergesi
- Link 0 : Port 0 aktif göstergesi
- Link 1 : Port 1 aktif göstergesidir.

2.4.2 Veri işleme arayüzü (PDI)

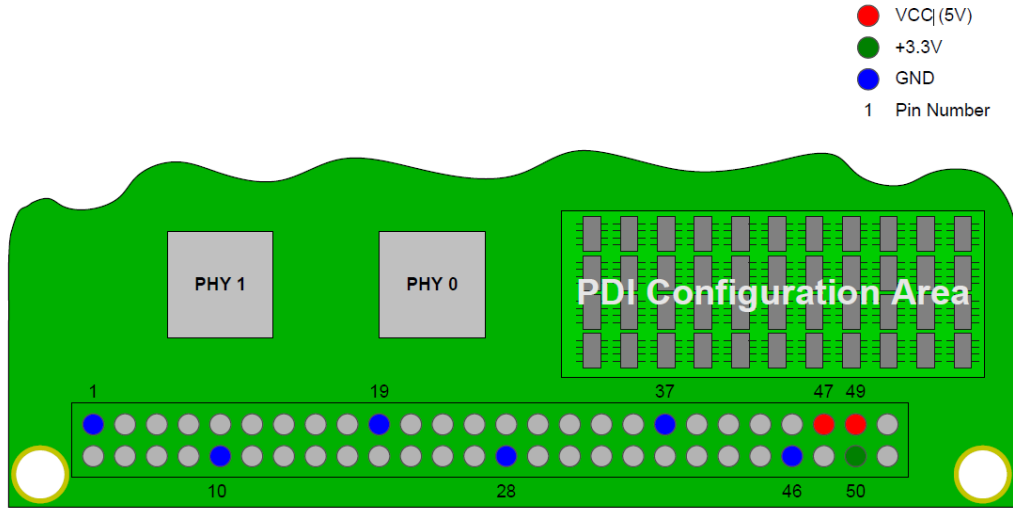
FB1111-14X kartının 3 çeşit konfigürasyonu vardır. Bunlar;

- FB1111-140 : Mikrokontrollör konfigürasyonu
- FB1111-141 : SPI konfigürasyonu
- FB1111-142 : Dijital IO konfigürasyonudur.

PDI harici donanım ile haberleşmek için kullanılır. FB1111-142 kartı veri işleme arayüzünün konfigürasyonunun değiştirilmesiyle FB1111-140 ve FB1111-141 kartına dönüşebilmektedir. EL9803 adaptörünün kullanılmasıyla PDI konfigürasyonlarında değişiklik yapılmadan FB1111-140 ve FB1111-141 kartına çevrilebilmektedir.

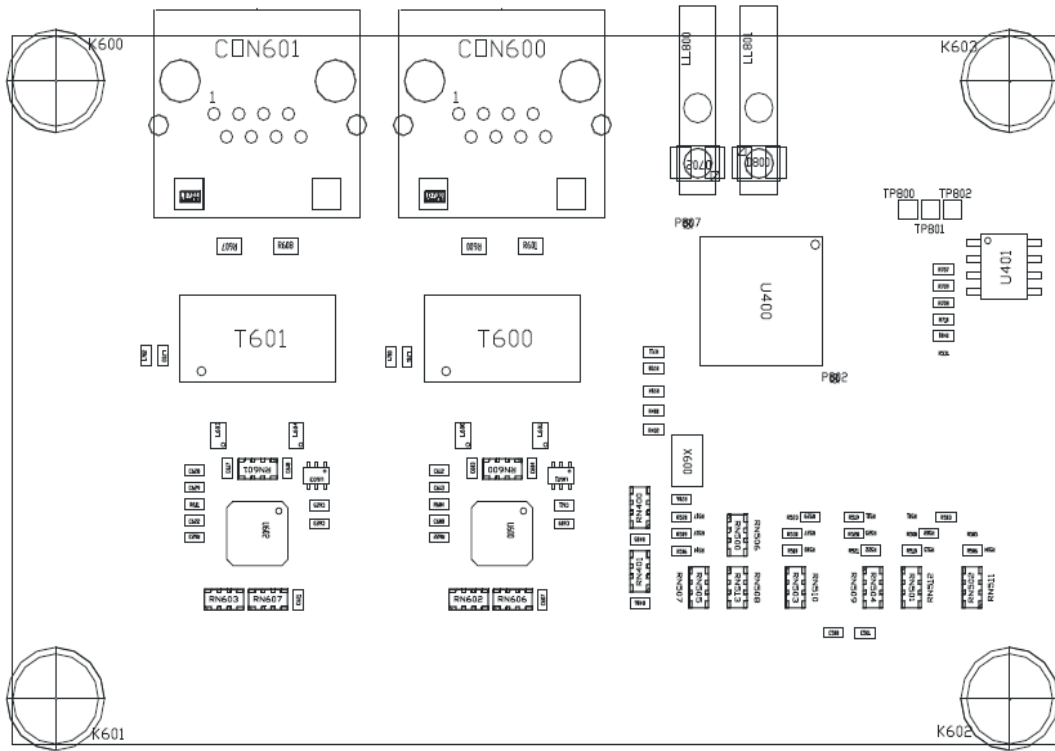
2.4.3 Güç kaynağı ve bağlantılar

Şekil 2.15’de FB1111 kartının pin numaraları gösterilmiştir. 47 ve 49. pinlere 5V uygulanmalıdır. 1, 10, 19, 28, 37, ve 46 numaralı pinlere GND bağlanmalıdır. 50. pin DC-DC konvertörün 3.3V çıkışıdır. 2 adet 5V girişi 2 farklı DC-DC konvertöre gitmektedir. Bir konvertör PHY 0 ve PHY 1 için güç sağlamaktadır. Diğer konvertör ise ET1100 slave kontrolcüsü ve kart üzerindeki diğer birimler için güç sağlamaktadır. 3.3V çıkışı (pin 50) 100mA ile sınırlandırılmıştır. Şekil 2.15 üzerinde mavi olarak işaretlenen pinler GND, kırmızı işaretli pinler 5V giriş ve yeşil işaretli pin ise 3.3V çıkıştır.

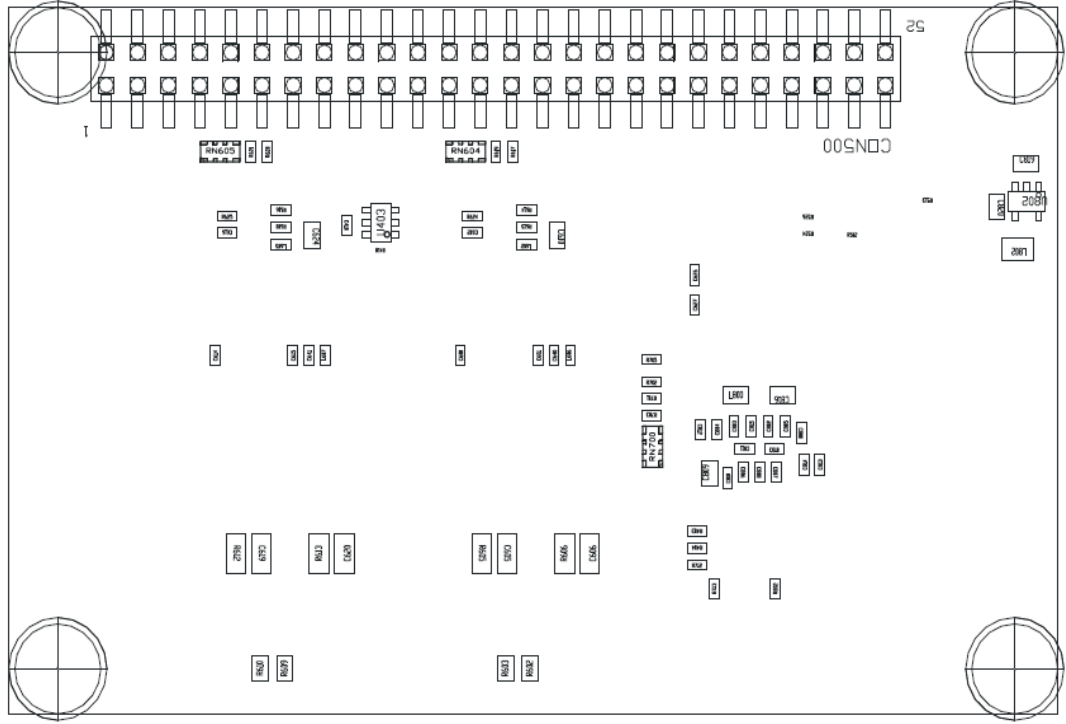


Şekil 2.15 : Pin numaraları ve güç girişleri

2.4.4 Komponent yerleşim görüntüsü



Şekil 2.16 : Üstten görünüş.



Şekil 2.17 : Alttan görünüş.

2.4.5 Güç test padleri

Çizelge 2.1’de gösterilen padler üzerinde güç testi yapılabilir.

Çizelge 2.1 : Test pedleri.

Gösterge	Sinyal	Bağlantısı
TP800	2.5V	ET1100
TP604	3.3V	3.3V
TP605	GND	GND

2.5 Arduino Due

Arduino Due, Atmel SAM3X8E ARM Cortex – M3 CPU’ya sahip bir mikrokontrolcüdür. İlk 32 bit ARM işlemci temelli Arduino devre kartıdır. 54 GPIO bulunur. Bunlardan 12 tanesi PWM çıkışı olarak kullanılabilir. 12 analog giriş, 2 UART, 84 MHZ saat sinyali, 2 DAC, 2 TWI bulunmaktadır. 6 – 20 Volt arası bir kaynak ile kart beslenebilmektedir. SAM3X kodları saklamak için 512 KB (2 x 256KB) hafızaya sahiptir. 64 KB ve 32 KB’dan oluşan 96 KB SRAM vardır (Şekil 2.18).

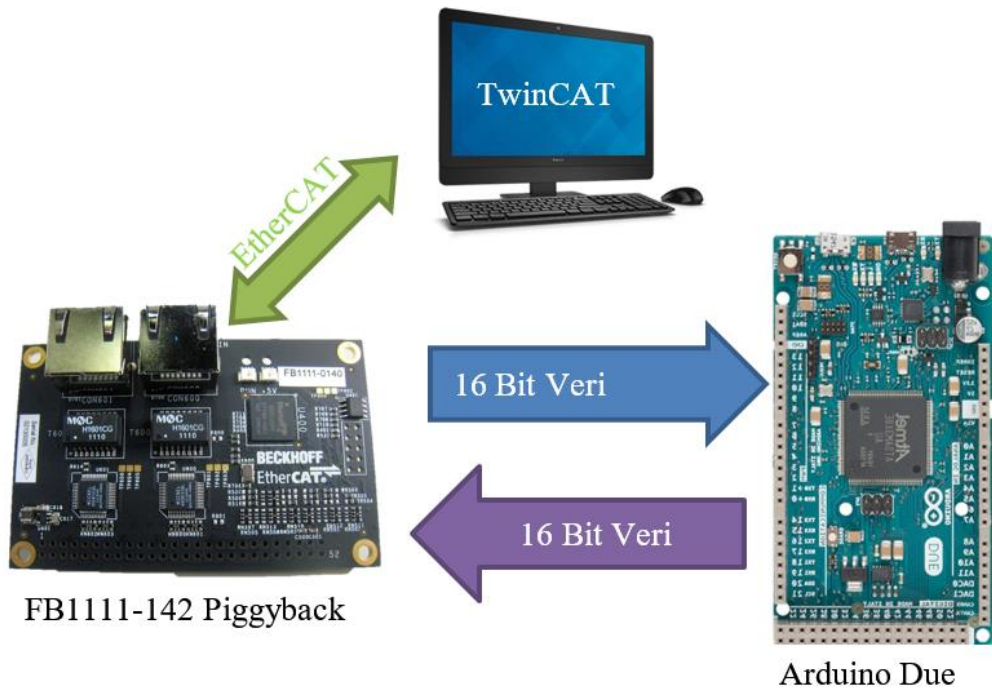


Şekil 2.18 : Arduino Due.

3. YAZILIM

3.1 Genel Yapı

Şekil 3.1’de sistemin genel haberleşme yapısı gösterilmiştir. Sistemde TwinCAT programı üzerinden gönderilen komutlar FB1111-142 kartına EtherCAT haberleşmesiyle iletilmektedir. FB1111-142 kartı aldığı komutları 16 bit halinde paralel olarak Arduino Due kartına iletmektedir. Arduino Due kartı ile FB1111-142 kartı arasında paralel olarak haberleşme sağlanmaktadır. FB1111-142 kartının veri iletimi esnasında Arduino Due kartı kendisine daha önce verilmiş olan komutun çıktılarını eş zamanlı olarak FB1111-142 kartına iletmektedir.



Şekil 3.1 : Genel haberleşme yapısı.

3.1.1 Veri yapısı

Şekil 3.1’de gösterildiği gibi FB1111-142 kartı ile Arduino Due arasında 16 bitlik giriş ve çıkışlık haberleşme yapısı bulunmaktadır. Bu 16 bitin 5 biti adres, 3 biti komut ve

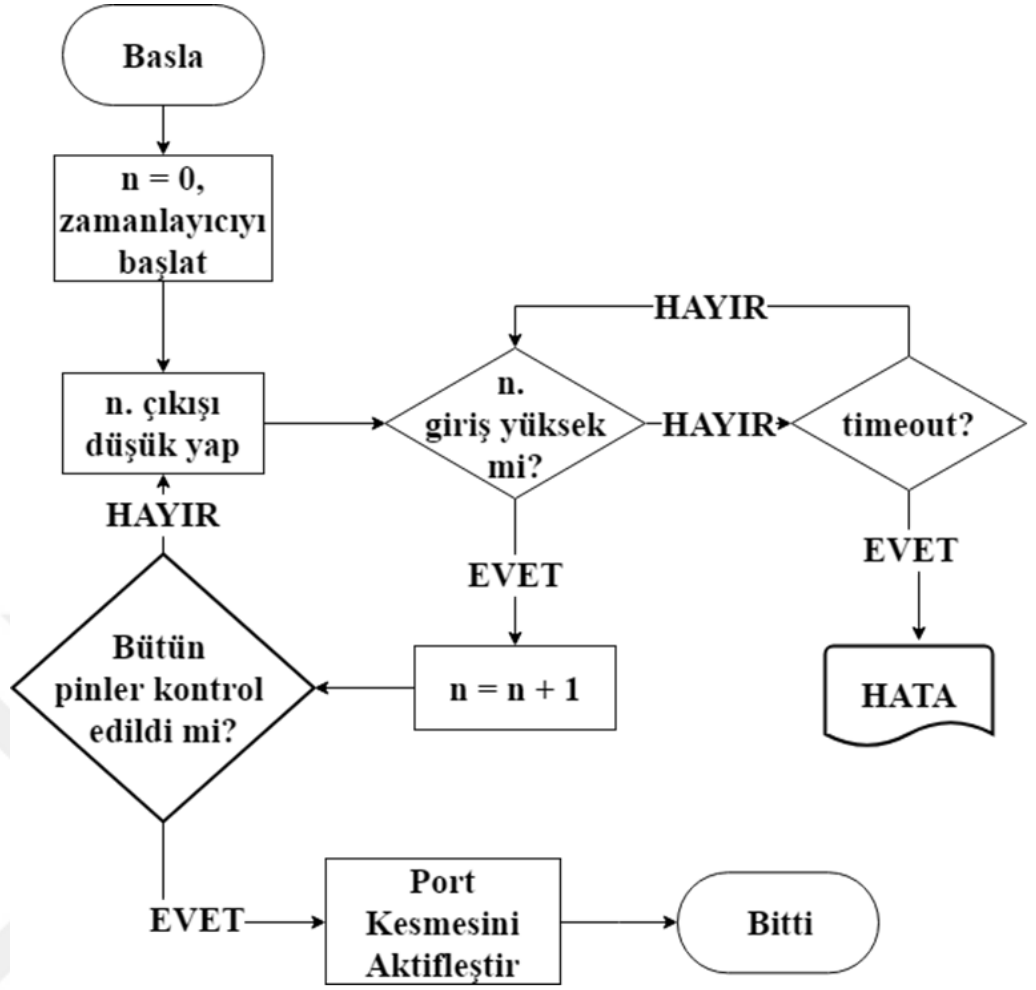
8 biti veri olarak ayarlanmıştır. Çizelge 3.1’de 3 bitlik komut satırlarının değerleri ve ilgili değerlerdeki işlemler gösterilmiştir. Arduino Due, FB111-142 tarafından verilen komutlara, yine aynı komut bitlerini kullanarak cevaplamaktadır.

Çizelge 3.1 : Komut bitleri ve anlamları.

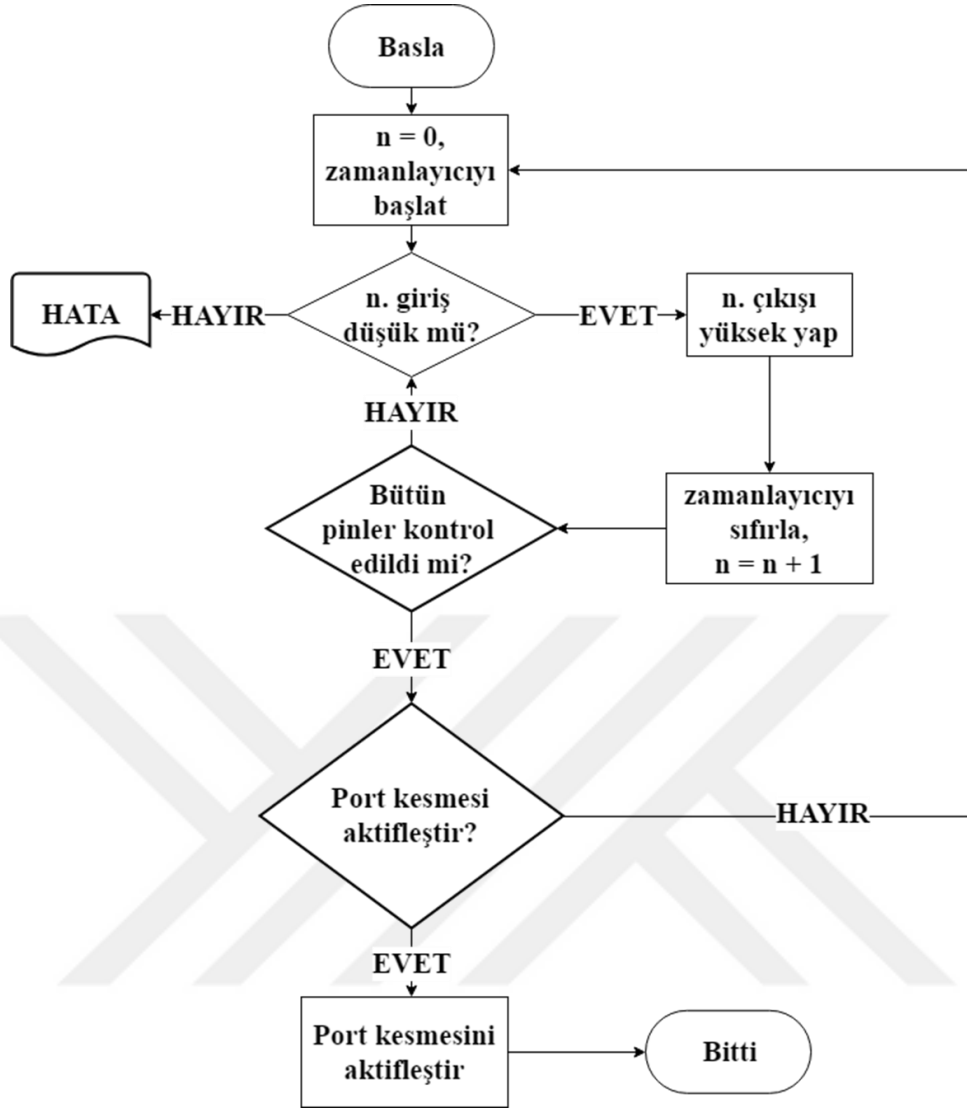
Komut	Kolon B
000	Operasyon Yok
001	ADC oku
010	PWM çıkışı
011	PTO çıkışı
100	ADC2 oku
101	Rezerve edilmiştir
110	Register Yaz
111	-

3.2 Başlama Algoritması

Haberleşmenin doğru bir şekilde sağlanabilmesi için bütün pinler bağlı olmalıdır. Bu sebeple başlama esnasında öncelikle bütün pinlerin bağlı olduğundan emin olunmalıdır. Bunu sağlamak için ‘baslama’ adımı TwinCAT üzerinden öncelikle 1. çıkış yükseğe çekilir. 1. çıkış yüksek olduktan sonra Arduino Due kartı kendi girişlerinden 1. girişin yüksek olduğunu algılar. Bunun üzerine Due kartı 1. çıkışını yüksek yapar. TwinCAT 1. girişin yüksek olduğunu algılayınca 2. çıkışını yüksek yapar. 16 çıkışın ve girişin karşılıklı tamamlanmasının ardından başlama adımı son bulur (Şekil 3.2) (Şekil 3.3).



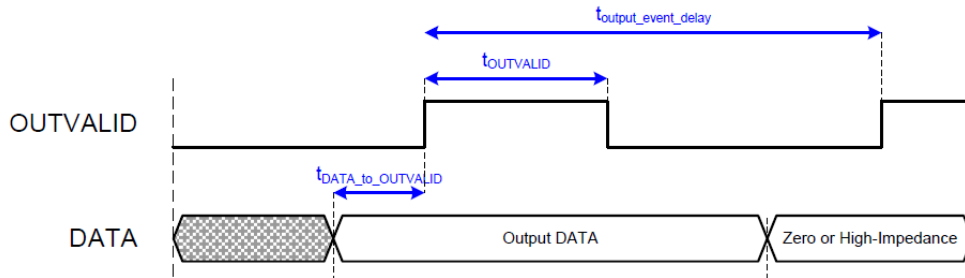
Şekil 3.2 : TwinCAT başlama akış şeması.



Şekil 3.3 : Arduino Due başlama akış şeması.

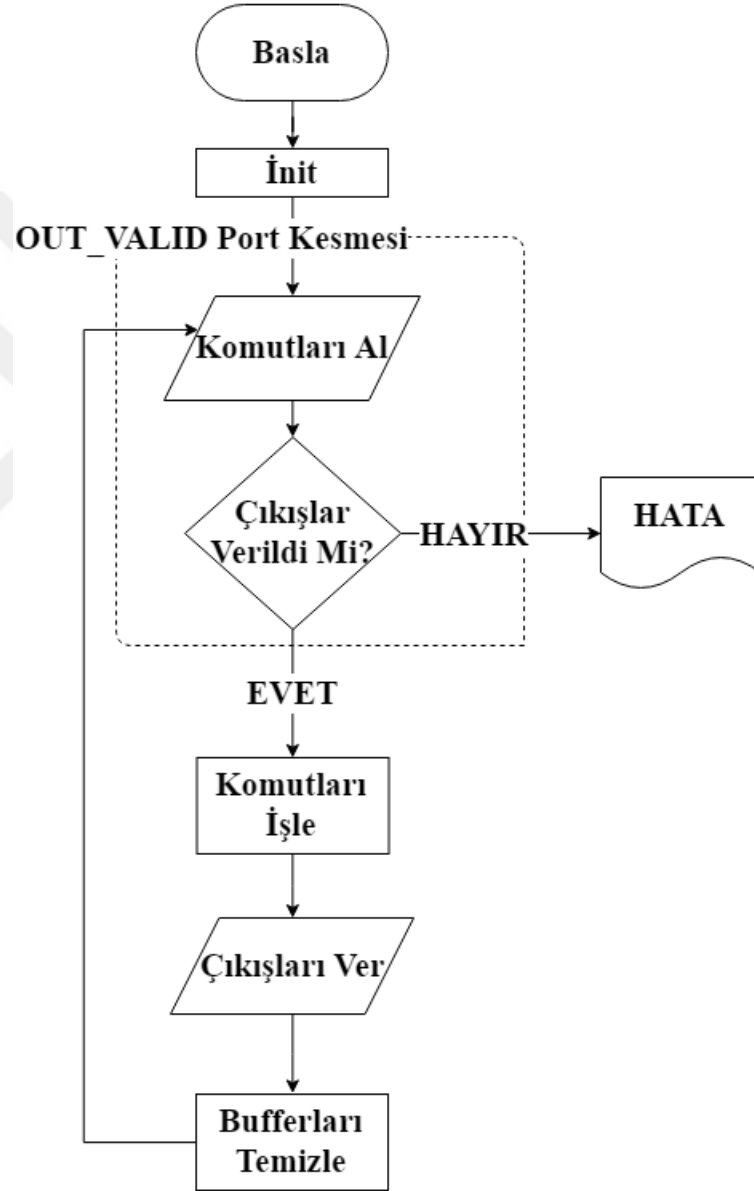
3.3 Döngüsel Çalışma

FB1111-142 kartında 'OUT_VALID' isimli bir çıkış bulunmaktadır. 38. pine bağlı olan bu çıkış, EtherCAT haberleşmesi sonunda FB1111-142 kartının çıkışlarının geçerli olduğuna dair bir pulse verir (Şekil 3.4).

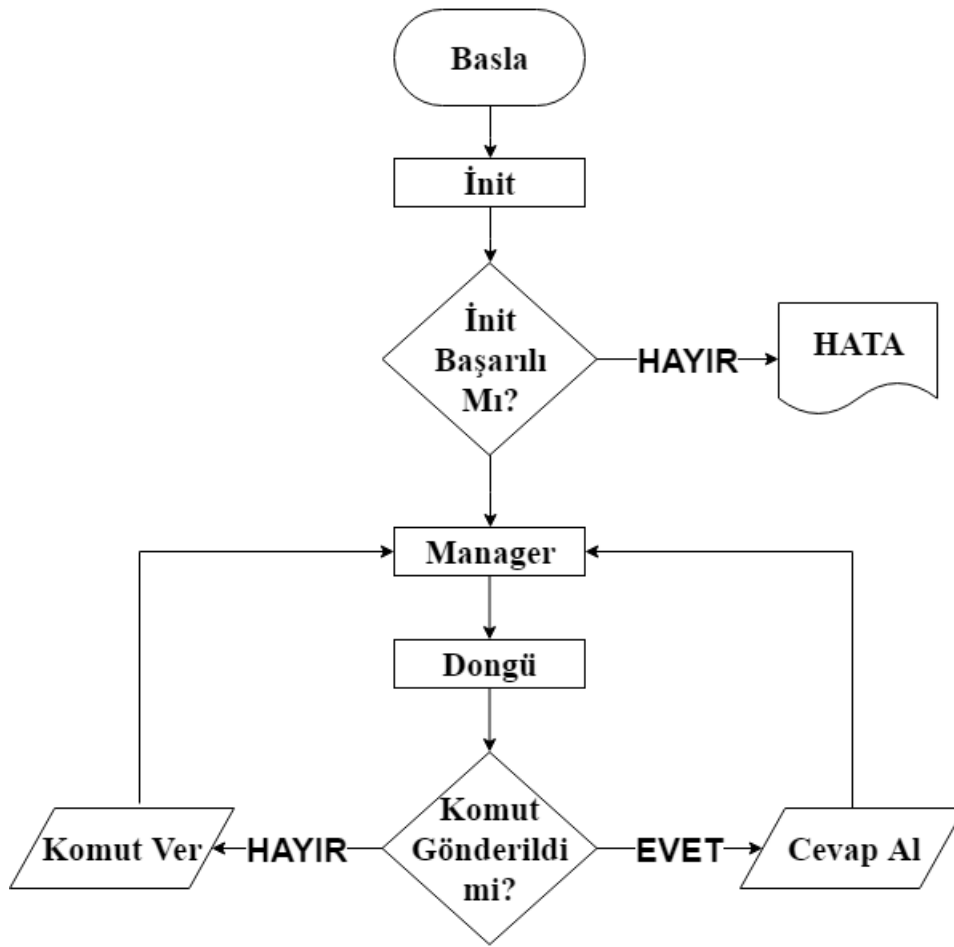


Şekil 3.4 : OUT_VALID zamanlaması [17].

Arduino Due üzerinde 8. pine port kesmesi atanarak, OUT_VALID olduğunda gerekli pin okumaları yapılmıştır. Aynı zamanda bir önceki çevrimde verilen komutların sonuçlarının pinlere verildiği kontrol edilmiştir. Çevrim süresi portların açılıp kapanma ve gerçek zaman çevrim süresiyle bağlantılı olup 1 ms(1KHz)'dir. Bütün işlemler 1 ms içinde bitecek şekilde ayarlanmıştır. Kurulan durum makinesi yapısıyla sistem kendi içerisinde döngüyü sağlamaktadır (Şekil 3.5).



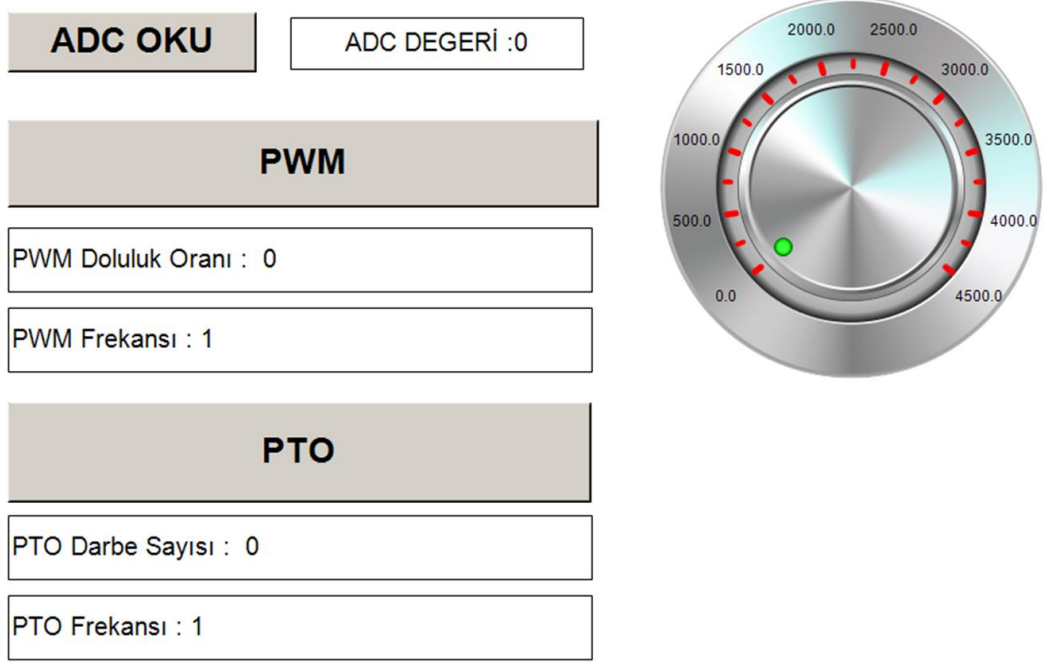
Şekil 3.5 : Arduino Due durum makinesi.



Şekil 3.6 : TwinCAT durum makinesi.

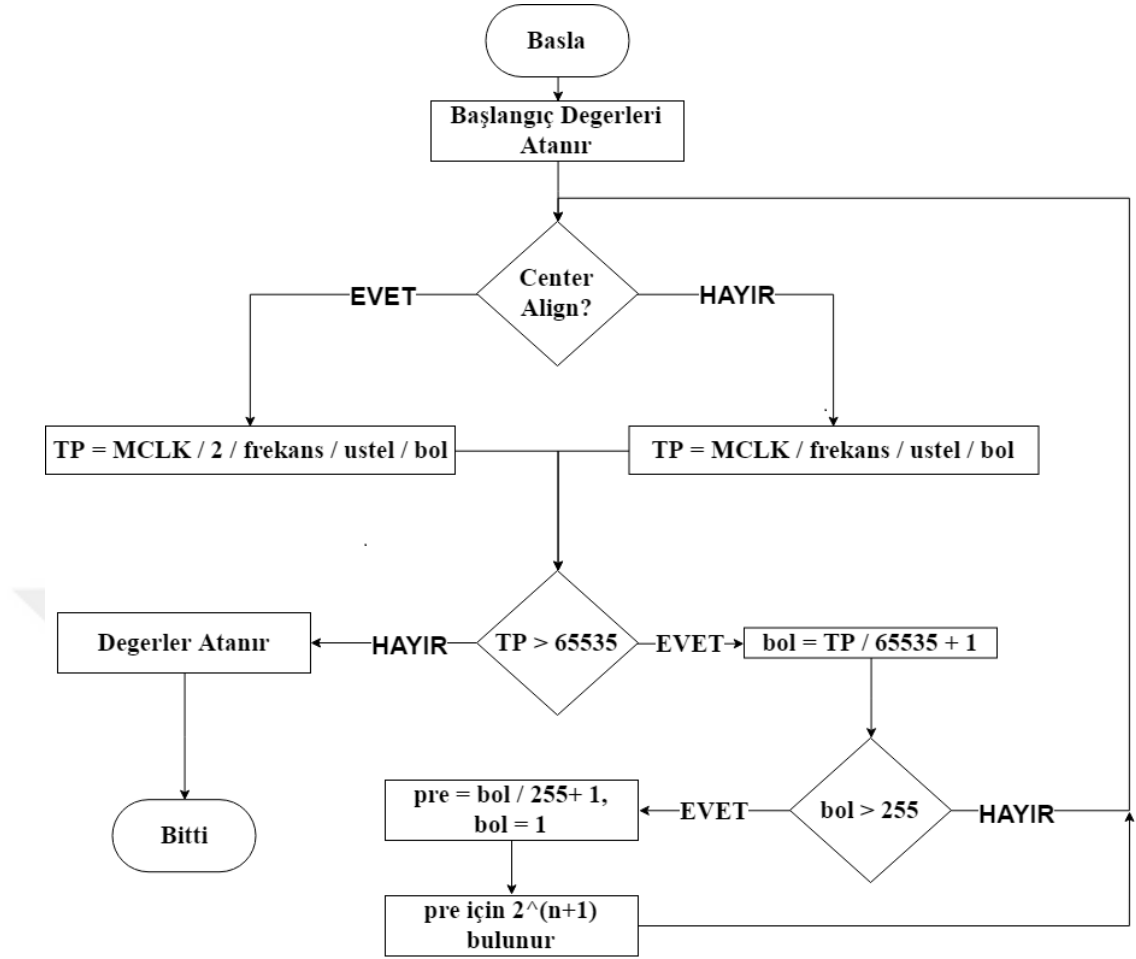
3.4 Arayüz

Sistem üzerinde ADC değerinin okunabildiği, PWM ve PTO hedef frekanslarının, PWM doluluk oranının, PTO pals sayılarının ayarlanabildiği bir arayüz bulunmaktadır (Şekil 3.7).



Şekil 3.7 : TwinCAT arayüzü.

Geliştirilen algoritma sayesinde, hedef frekans değerine göre periyot registerının alabileceği en yüksek değeri alarak, bölme ve frekans sayıcı (prescaler) değerleri otomatik olarak hesaplanmaktadır (Şekil 3.8). Böylece arayüzde sadelik sağlanmıştır.

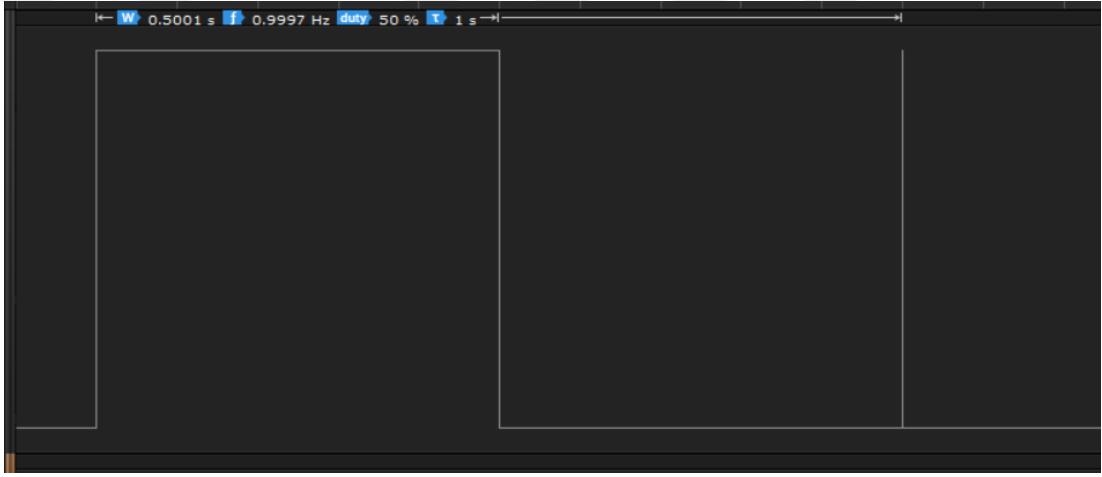


Şekil 3.8 : Frekans hesaplama algoritması.

4. SONUÇ VE ÖNERİLER

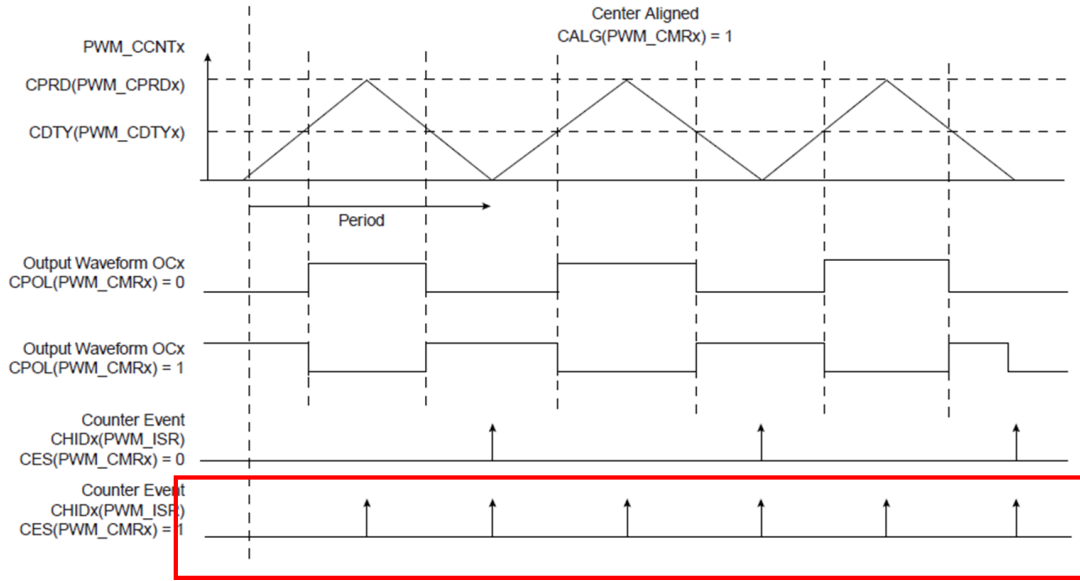
Arduino Due üzerinde çevrimlerin yetişmemesi sebebiyle Arduino kütüphanesinin hazır komutları yerine register bazlı olarak kodlama yapılmıştır. Böylece çevrimler yakalanabilmiştir.

PTO çıkışı esnasında, tek pals komutu verildiğinde istenmeyen bir ‘glitch’ hata gözlemlenmiştir (Şekil 4.1).

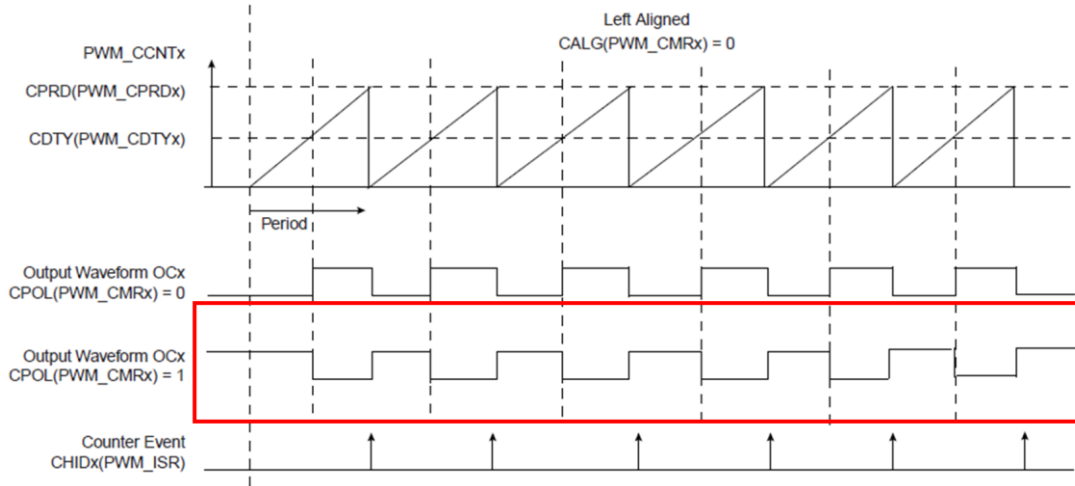


Şekil 4.1: İstenmeyen hata.

Hatanın giderilmesi için öncelikle PWM - PTO kesmesi içerisinde, kesmeye girer girmez çıkışın sıfırlanması denenmiştir. Ancak sorun giderilememiştir. Ardından PWM – PTO sinyali merkez hizalamalı çalıştırılarak kesme sinyalinin oluşturulan sinyal periyodunun ortasında ve sonunda alınması sağlanmıştır (Şekil 4.2). Sorun giderilememiştir. Ancak PWM – PTO sinyallerini sol hizalı olarak ayarlayarak (Şekil 4.3) ve polaritesini değiştirerek istenen sinyal hatasız olarak elde edilmiştir (Şekil 4.4). ADC ölçümü 12 bit olarak ölçülerek TwinCAT arayüzüne aktarılmıştır. Frekans ayarlamaları başarılı bir şekilde register komutu ile yapılmıştır.



Şekil 4.2 : Merkez hizalı PWM - PTO sinyali kesmesi [17].



Şekil 4.3 : Sola hizalama ve polaritenin değiştirilmesi [17].



Şekil 4.4: 500KHz'de 5 adet puls verilmesi.

KAYNAKLAR

- [1] **Url-1** <<http://164.100.133.129:81/econtent/Uploads/MAS-2%20Automotive%20sensors%20and%20actuators.pdf>>, erişim tarihi 04.04.2019.
- [2] **Url-2** <<https://www.ece.uvic.ca/~cai/can.pdf>>, erişim tarihi 10.04.2019.
- [3] **Url-3** <<https://www.ethercat.org/en/technology.html>>, 13 04 2019.
- [4] **G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano ve C. Zunino** (2010). On the Accuracy of the Distributed Clock Mechanism in EtherCAT. *IEEE International Workshop on Factory Communication Systems Proceedings*, Torino.
- [5] **J. Robert, J.-P. Georges, E. Rondeau, T. Divoux** (2012). Minimum cycle time analysis of Ethernet-based real-time protocol. *International Journal of Computers, Communications and Control*, pp. 743-757, Nancy, France.
- [6] **R. Schlesinger, A. Springer ve T. Sauter** (2014). New approach for improvements and comparison of high performance real-time ethernet networks. *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, Spain.
- [7] **D. Orfanus, R. Indergaard, G. Prytz ve T. Wien** (2013). EtherCAT-based platform for distributed control in high-performance industrial applications. *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Cagliari, Italy.
- [8] **K. Langlois, T. v. d. Hoevena, D. R. Ciancaa, T. Verstratena, T. Baceka ve Bryan** (2018). EtherCAT Tutorial: An Introduction for Real-Time Hardware Communication on Windows [Tutorial]. *IEEE Robotics & Automation Magazine*, cilt 25, no. 1, pp. 22-122.
- [9] **I.-K. Jung ve S. Lim** (2012). An EtherCAT based Real-time Centralized Soft Robot Motion Controller. *International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA)*, Sanya, China.
- [10] **Dafa Shi** (2012). Wind Power System Main Controller Based on EtherCAT. *IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*, Kuala Lumpur, Malaysia.
- [11] **Url-4** <<https://www.wikiwand.com/tr/Bilgisayar>>, erişim tarihi 15.04.2019.

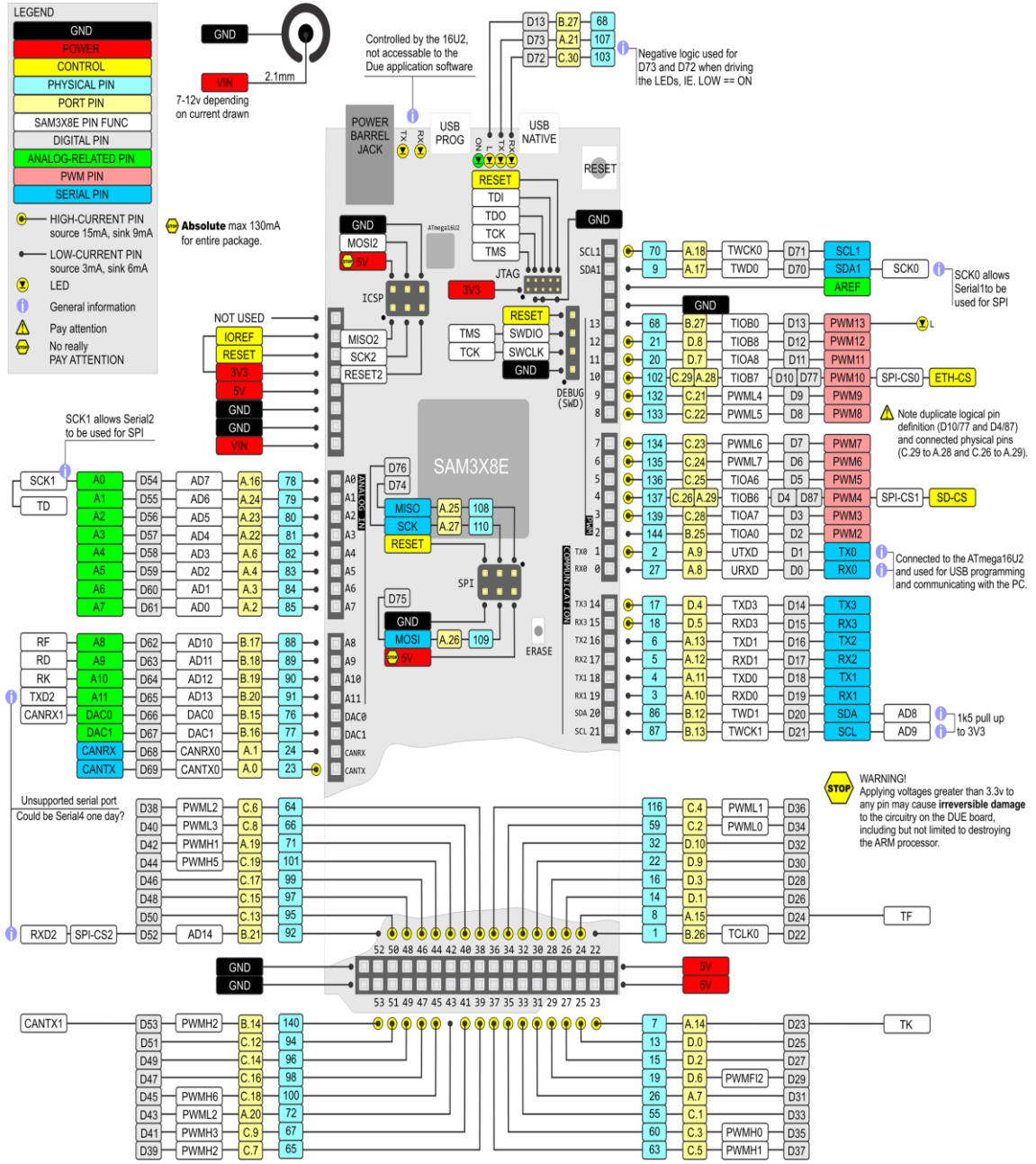
- [12] **H. Kopetz** (1997). Real-Time Systems: Design Principles for Distributed Embedded Applications, London: Springer.
- [13] **R. Barth** (2011). Real-time processing – the basis for PC Control. Eriřim: 15 Nisan 2019, http://www.pc-control.net/pdf/special_25_years_pcc/products/pcc_special_0811_realtime_e.pdf.
- [14] **Beckhoff řirketi** (2019). Eriřim: 15 Nisan 2019, https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_system/html/TcSysMgr_SystemNode_SubNodes_RealTime.htm&id=.
- [15] **Url-5** <<https://www.wikiwand.com/en/Arduino>>, eriřim tarihi 16.04.2019.
- [16] **Url-6** <https://download.beckhoff.com/download/Document/io/ethercat-development-products/beckhoff_fb1111-014x_v22.pdf>, eriřim tarihi 16.04.2019.
- [17] *Hardware Data Sheet ET1100*, Beckhoff Company.

EKLER

EK A: Arduino Due pin tablosu.



EK A



ÖZGEÇMİŞ

Ad - Soyad : Ömer Faruk ŞAHAN
Doğum Tarihi ve Yeri : 1992, Kahramanmaraş
E-posta : omerf.sahan@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2016, Yıldız Teknik Üniversitesi, Makine Fakültesi, Mekatronik Mühendisliği