

AIRHMI LCD EKCRAN EDITOR KILAVUZU



AirHMI LCD EKCRAN EDITÖR KILAVUZU

AIRHMI LCD EKLAN EDITOR KILAVUZU

AirHMI Visual Screen Creator, AirHMI LCD ekranları için İnsan Makine Arayüzü GUI'lerini tasarım açışından en üst seviyede memnuniyet ve en verimli sürede oluşturabilmek amacıyla tasarlanmıştır. Editör kullanımında Tasarım ve Programlama dünyasına ait işlevselliklerimiz bulunmaktadır: Görsellik açışından zengin nesne hazinesinden özgün olabileceğiniz ve istekleriniz doğrultusunda rahatlıkla oluşturabileceğiniz ekran tasarımı desteğinin yanı sıra programlama kısmında da kullanıcıya birçok kolaylık sağlamaktadır.

AIRHMI

AİRHMI LCD EKРАН EDITOR KILAVUZU

İÇİNDEKİLER

1.	AirHMI Visual Screen Creator KURULUMU	1
2.	PROJE OLUŞTURMA	2
3.	CİHAZ BAĞLANTISI	4
4.	AirHMI EDİTOR ANA ARAYÜZÜ.....	5
4.1	BAŞLIK ÇUBUĞU.....	5
4.2	ANA MENÜ ve ARAÇ ÇUBUKLARI	5
4.3	BİLEŞENLER BÖLMESİ	7
4.4	EKRAN / KOMUT SEKMESİ	8
4.5	TASARIM ANA EKРАН ALANI	9
4.6	GÖRSELİ OLMAYAN BİLEŞENLERİN ALANI.....	10
4.7	NESNELERİN ÖZNİTELİK ALANI.....	10
4.8	3.7.1 Projede Kullanılan Nesnelerin Gösterim Alanı.....	10
4.9	3.7.2 Nesnelerin Öznitelikleri Gösterim / Ayar Alanı	11
4.10	ÖZNİTELİKLERİN AÇIKLAMA ALANI.....	11
4.11	KULLANICI PROJE KODU MENÜ ve ARAÇ ÇUBUKLAR	11
4.12	KULLANICI PROJE KOD ALANI	11

AİRHMI LCD EKLAN EDITOR KILAVUZU

4.13	KOD ALANI ZOOM ALANI.....	12
4.14	KOD ALANI.....	12
5.	AİRHMI NESNELERİ VE FONKSİYONLAR.....	14
5.1	TIMER	14
5.2	Button	17
5.3	Label.....	23
5.4	Image.....	28
5.5	ProgressBar	33
5.6	Slider	38
5.7	Gauge	43
5.8	VARIABLE.....	48
5.9	Delay().....	56
5.10	uartDataGet ().....	57
5.11	ChangeScreenSet ().....	58
5.12	dateSet ()	59
5.13	timeSet ().....	60
5.14	dateGet ().....	61

AIRHMI LCD EKCRAN EDITOR KILAVUZU

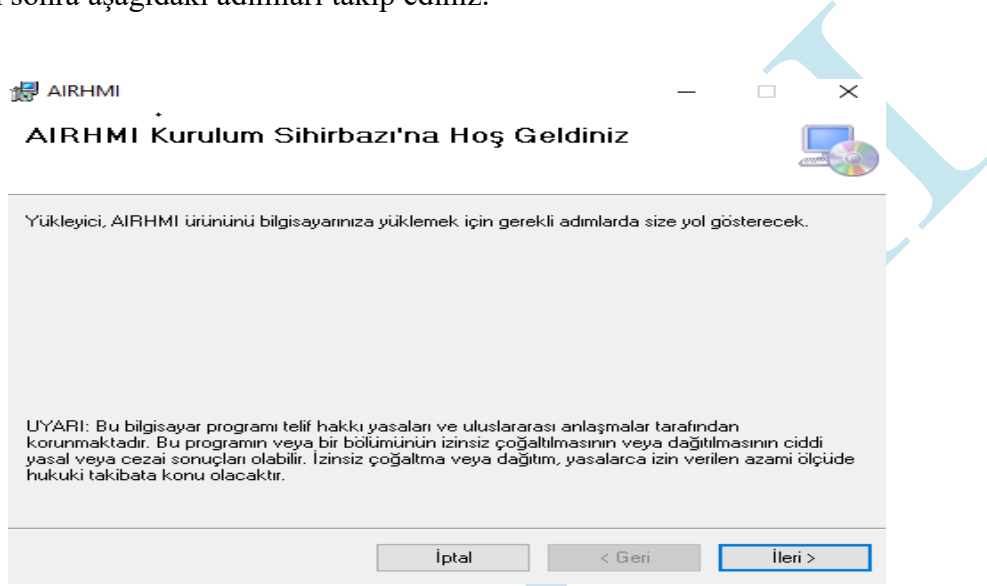
5.15	timeGet ()	62
5.16	AudioPlay()	62
5.17	AudioStop()	63
5.18	AudioStatusGet()	64
5.19	File_write ()	65
5.20	File_read()	66
5.21	File_size()	67
5.22	GPIO_Write()	68
5.23	GPIO_Read()	69
5.24	PWM_Set()	70
5.25	BuzzerSet()	71

AIRHMI LCD EKРАН EDITOR KILAVUZU

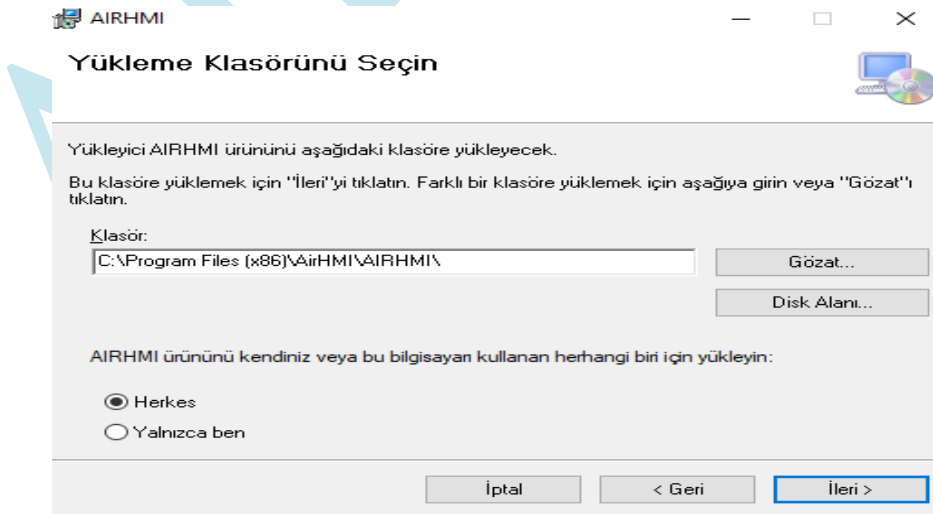
1. AirHMI Visual Screen Creator KURULUMU

İndirme Linki: <https://www.airhmi.com/airhmi-visualcreator>

AirHMI Editör'ü bilgisayarınıza yüklemek için AIRHMISETUP.msi dosyasına çift tıklayın. Bu işlemten sonra aşağıdaki adımları takip ediniz.



Yükleme klasörünü ve diğer seçenekleri istediğiniz şekilde seçip ileri tuşuna basarak yükleme başlatılır.

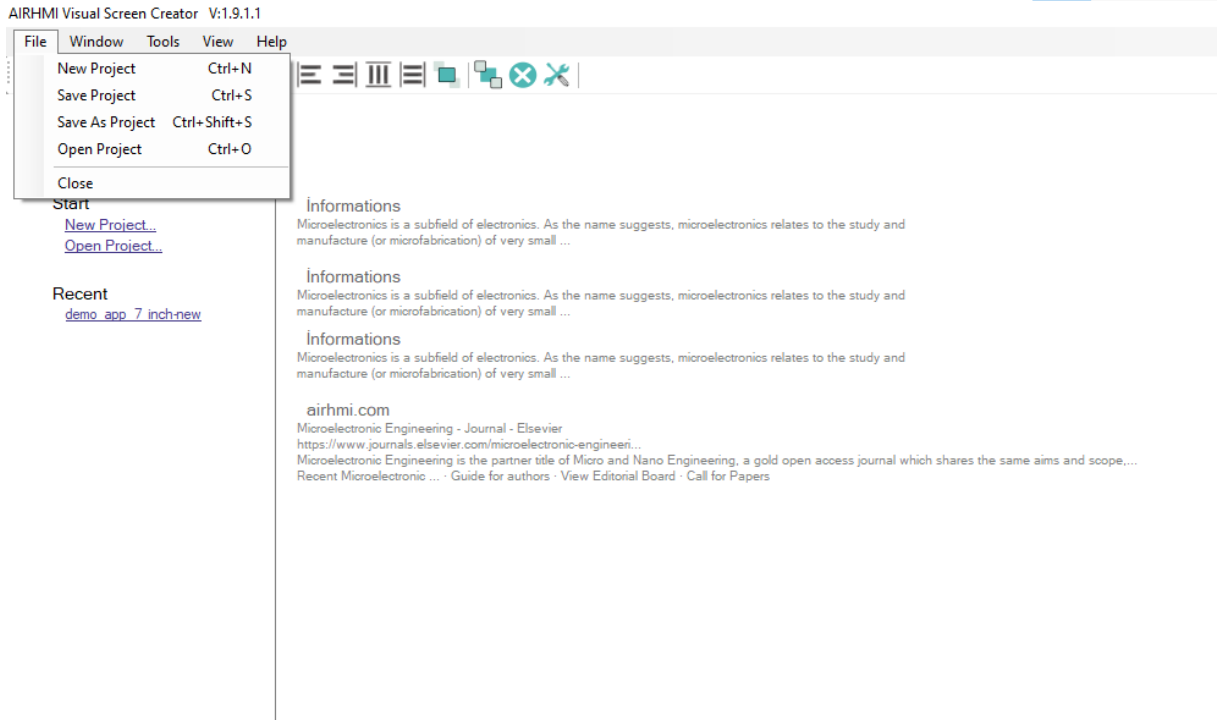


AIRHMI LCD EKLAN EDITÖR KILAVUZU

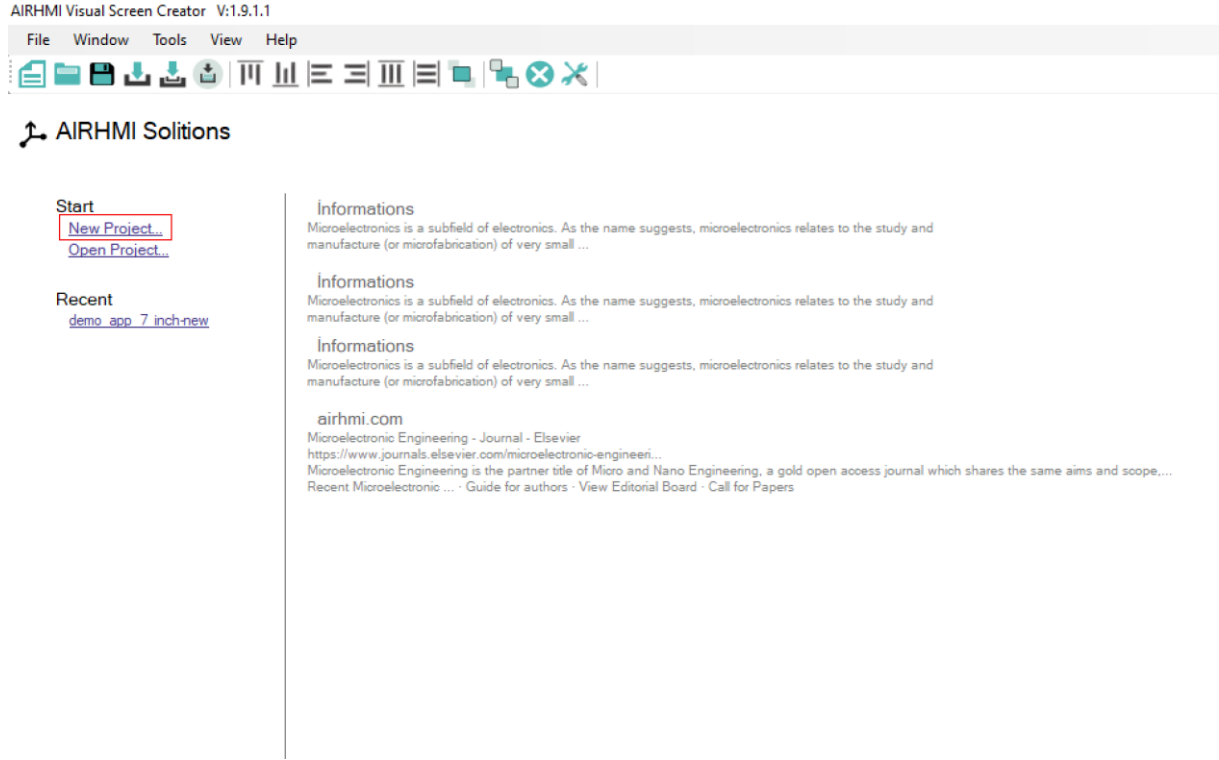
2. PROJE OLUŞTURMA

AirHMI ile arayüz oluşturmak için öncelikle AirHMI Editör programını indirip bilgisayarınıza kurmanız gerekmektedir. AirHMI Editör programındaki sürükle-bırak özelliği arayüz geliştirmeyi kolaylaştırmaktadır. AirHMI Editörü ile projelerinize, Buton, Resim, Yazı, İlerleme çubuğu, Gauge, Key, Analog ve Dijital değerleri görmek için sayısal giriş ve çıkışlar gibi birçok bileşen ekleyebilirsiniz.

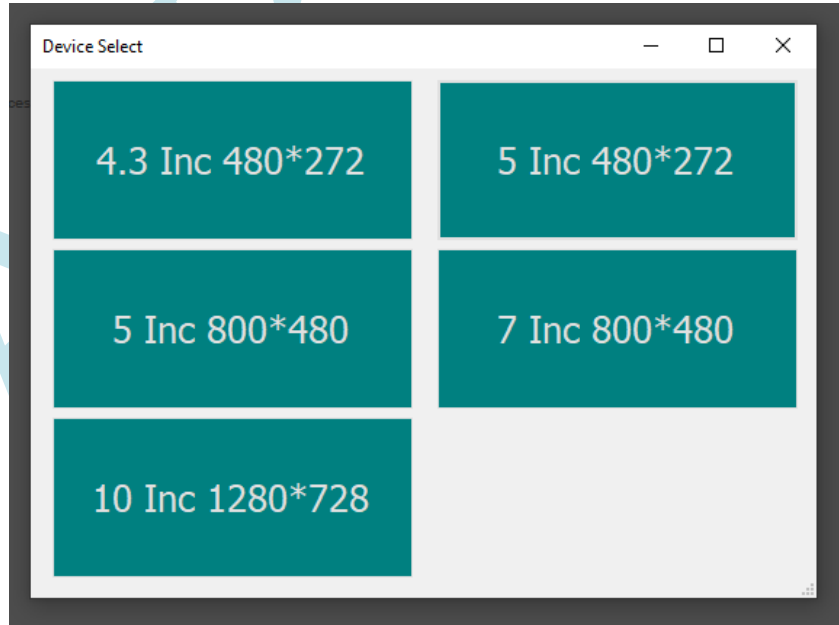
Programın kurulumu oldukça kolaydır. Kurulumu yaptıktan sonra AirHMI Editör programı çalıştırılmalıdır. Karşınıza aşağıdaki resimlerde görüldüğü gibi bir sayfa çıkacaktır. Bu sayfadan sol üst köşede bulunan File – New yolunu izleyerek veya programın ilk açılış sayfasında karşınıza çıkan sekmelerden New Project’e tıklayarak projenizi oluşturuyorsunuz.



AIRHMI LCD EKРАН EDITOR KILAVUZU



Kayıt işleminden sonra karşınıza aşağıdaki resimde görüldüğü gibi bir sayfa çıkacaktır. Karşınıza çıkan sayfada Ekrana ait boyut ve çözünürlük ile ilgili ayarlar yapılmalıdır.

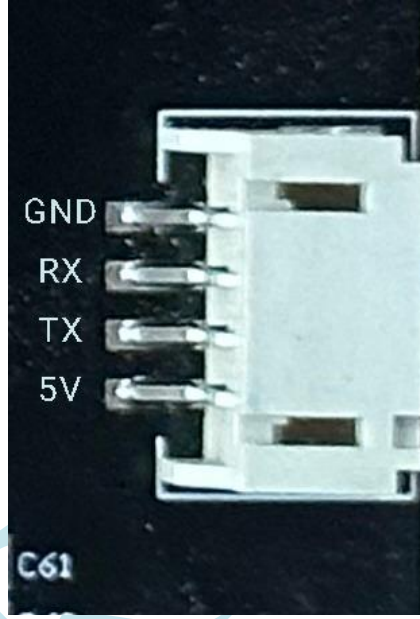


AIRHMI LCD EKРАН EDITOR KILAVUZU

3. CİHAZ BAĞLANTISI

AirHMI ekrana enerji verdiğimiz power konektör dört pinlidir. 1 ve 4 besleme , orta iki pin ise uart haberleşme pinleridir.

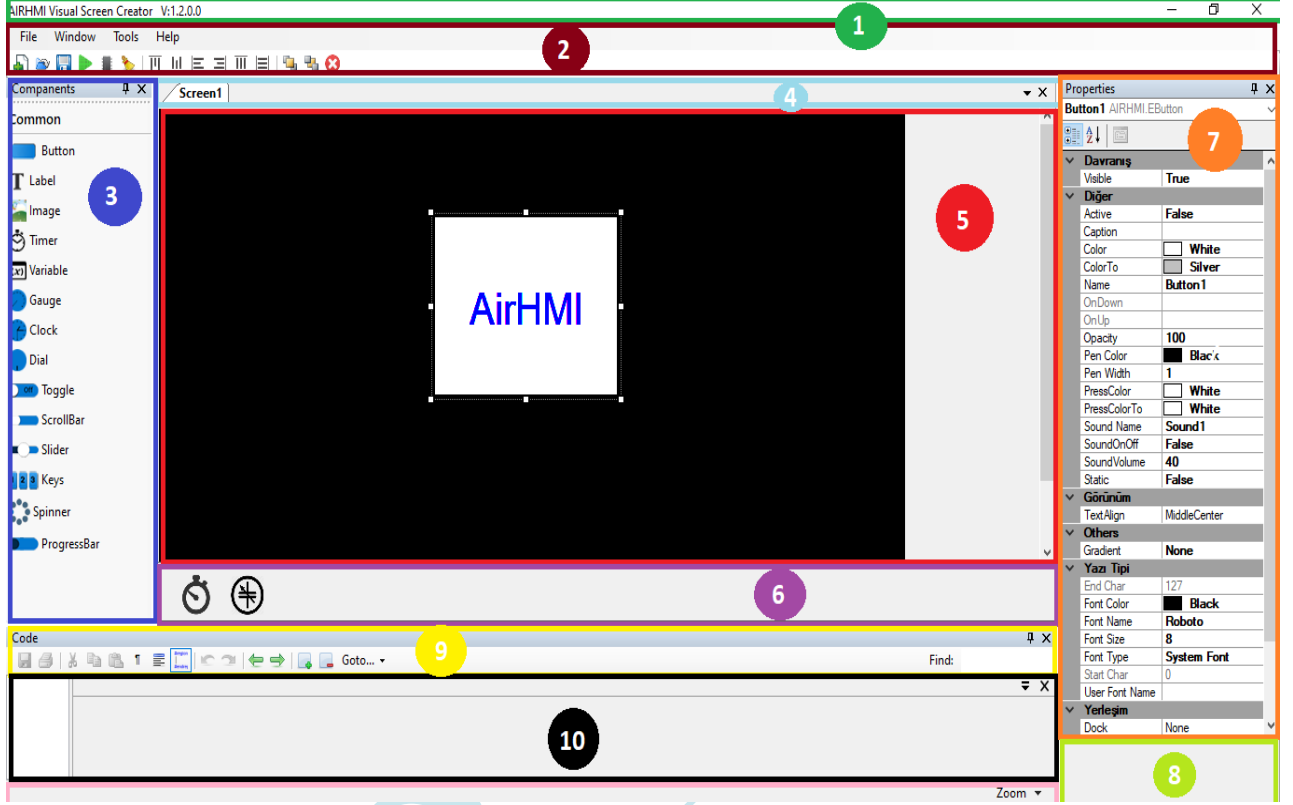
1) POWER konektöre ait pinler şu şekildedir;



Uyarı: 5V beslemeyi ters vermeyiniz. Beslemeyi ters vermeniz durumunda ekranınız zarar görebilir.

AIRHMI LCD EKРАН EDITOR KILAVUZU

4. AirHMI EDITOR ANA ARAYÜZÜ



4.1 BAŞLIK ÇUBUĞU

Başlık Çubuğu, bir AirHMI projesi açıldığında uygulama ismini ve versiyon numarasını içerir.

4.2 ANA MENÜ ve ARAÇ ÇUBUKLARI



Dosya (File) Menüsü

Kullanıcılar için Yeni Proje Açın, Projeyi Kaydet, Projeyi Farklı Kaydet, Var Olan Bir Projeyi Açın ve Çıkış gibi komutlar bulunmaktadır. Burada önemli olan nokta var olan bir proje

AİRHMI LCD EKРАН EDITOR KILAVUZU

açıkken yeni proje açmak istenildiğinde eski projenin bilgisayarda saklanması ya da yapılan değişikliklerin kaybolmaması isteniyorsa ekrana gelen kaydet mesajına onay verilmelidir.

Pencere (Window)

Pencere alanı içerisinde ;

- Projede kullanılan ana ekrana ek yeni çalışma ekranı oluşturma (Add Screen)
- Tasarlanan arayüz ekranının seçili USB port üzerinden AirHMI LCD Kartına yüklenmesi (Download to Flash)
- Tasarlanan arayüz ekranının harici dosyalar halinde bilgisayar içerisinde istenilen bir dosyaya çıkartılması (Download to SD Kart). USB yüklemenin istenmediği durumlarda SD Kart üzerinden Bootloader yükleme yapmak için kullanılmaktadır. Dosyalar SD karta kopyalanıp proje SD Kart üzerinden çalıştırıldığında dosyalar USB üzerinden yüklenir gibi SD Kart'tan yüklenmektedir.

Araçlar (Tools)

Araçlar içerisinde Options içerisinde USB yükleme için port seçme ve baud rate ayarlama bölümü bulunmaktadır. USB yükleme birçok baud rate değerinde çalıştığı için kullanıcı istediği baud rate ayarını seçerek yüklemesini gerçekleştirebilmektedir.

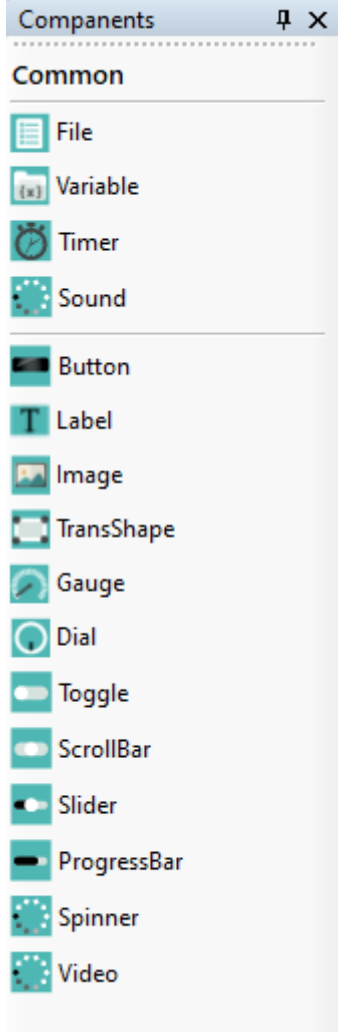
Hizalama



Sola Hizala, Sağa Hizala, Üst Hizala ve Alta Hizala; dikey ve yatay olarak ortalama özellikleri sayesinde belirlenen nesnelere istenen şekilde hizalanmış veya ortalanmış hale getirilir.

Öne Getir ve Arkaya Gönder özellikleri sayesinde iç içe geçen nesnelere hangisinin önde duracağı belirlenebilir ve arka planda durması istenen nesnelere için kullanılır.

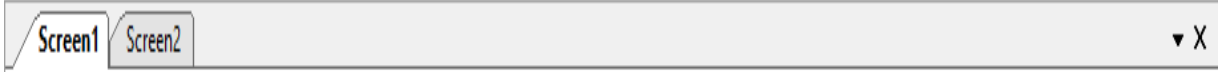
4.3 BİLEŞENLER BÖLMESİ



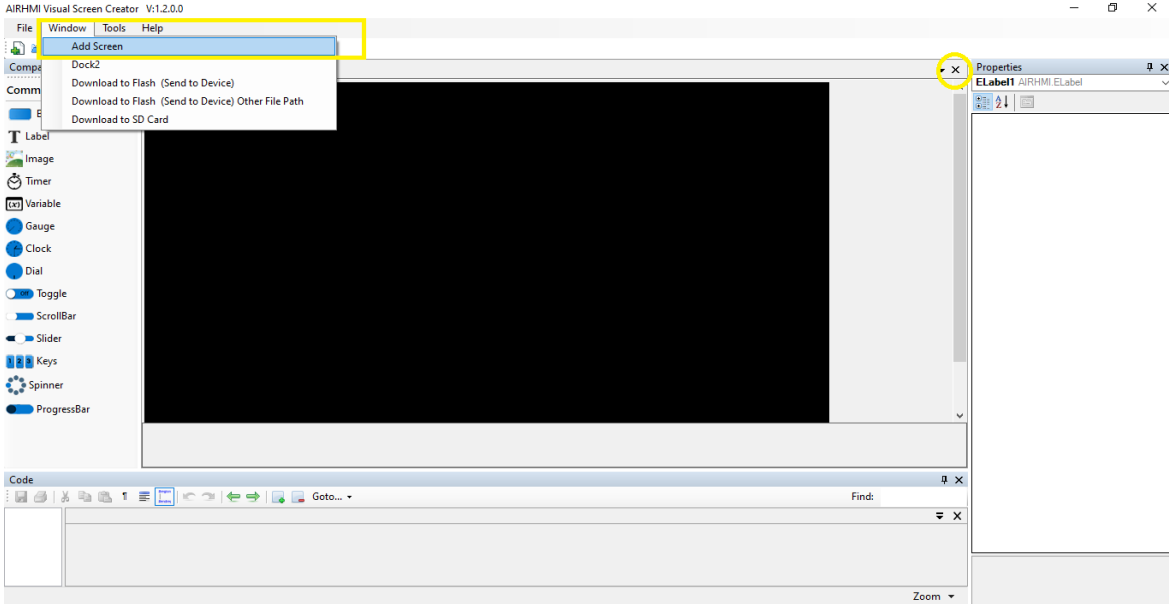
AirHMI LCD Tasarım Ekranı'nda gösterilecek hazır nesnelerin bulunduğu bölümdür. Kullanılmak istenilen nesne üzerine tıklanıp ekran alanına sürüklenerek projeye eklenmektedir. Ekranda gösterilmeyen harici nesneler de bu bölümde bulunmaktadır: Timer ve Variable. Bu nesneler ekran alanının alt kısmında Görseli Olmayan Bileşenlerin Alanı bölümünde bulunmaktadır. Tasarlanan proje özelinde nesnelerin özelliklerini (konumu, boyutu, ismi, vb...) ayarlama Nesnelerin Öznitelik Alanı adlı bölümde bulunmaktadır.

AIRHMI LCD EKРАН EDITOR KILAVUZU

4.4 EKРАН / KOMUT SEKМESİ



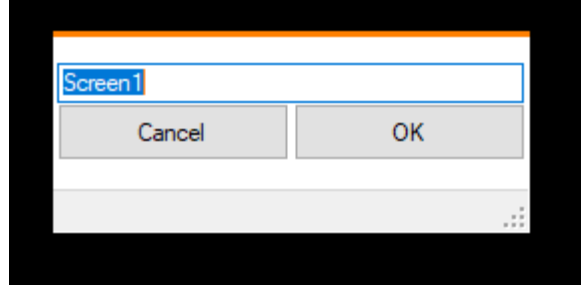
Tasarım projeleri genelde tek ekran olarak kullanılmayıp aynı anda farklı ekranlara ihtiyaç duymaktadır. Açılış Genel Gösterim Ekranı, Menü Ayar Ekranı, Detaylı Gösterim Ekranı vs... Bu nedenle AirHMI Editör içerisinde kullanıcı istekleri doğrultusunda birden fazla özgün ve yaratıcı ekran tasarımı yapabilmektedir. Ekran / Komut Sekmesi ile hangi ekranda çalışma yapılacağını seçme işlemi gerçekleştirilmektedir.



Yeni çalışma ekranı eklemek için Window/Add Screen sekmesi kullanılabilir veya çalışma sayfası üzerinde boş bir yerde sağ tıklanarak Add Screen seçilebilir. Açılmış olan çalışma sayfasını silmek için Ekran / Komut Sekmesi satırının sonunda yer alan çarpı(x) işaretine basmak yeterli olacaktır.

Ekranın ismini değiştirmek için ekranda boş bir alanda sağ tıklayarak Rename sekmesine tıklanır. Açılan sekmeden ekranın ismi değiştirilebilir.

AIRHMI LCD EKРАН EDITOR KILAVUZU



4.5 TASARIM ANA EKРАН ALANI

AIR HMI Designer çalışma ekranı tasarım görseli alanıdır. LCD Ekran tasarımında hangi nesnelerin ekranda nerede bulunacağı, boyutları, yazı özellikleri gibi özellikler bu alanda gösterilmektedir.

AHMI SCREEN EDITOR

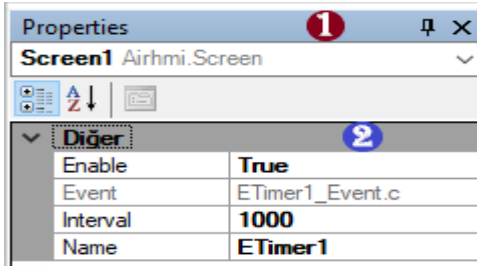
AİRHMI LCD EKРАН EDITOR KILAVUZU

4.6 GÖRSELİ OLMAYAN BİLEŞENLERİN ALANI



Hazırlanan bir projede bileşenlerin hepsi LCD ekranda gösterilmemektedir. Arka planda çok önemli görevlerde yer alırken LCD ekran üzerinde gösterilmesine gerek olmayan bileşenler de mevcuttur: Timer ve Variable gibi. LCD ekranda gösterilmeyen fakat tasarım esnasında kullanım kolaylığı sağlayabilmesi ve anlaşılabilir olabilmesi için arka planda çalışan bileşenlerin Editör içerisinde gösterilmesi önemlidir. Görseli Olmayan Bileşenlerin Alanı bu doğrultuda projede kullanılan Timer ve Variable gibi bileşenlerin gösterildiği alandır.

4.7 NESNELERİN ÖZNİTELİK ALANI



4.8 3.7.1 Projede Kullanılan Nesnelerin Gösterim Alanı

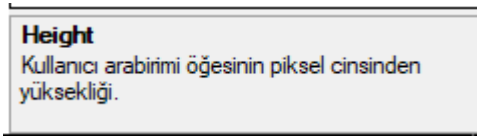
LCD ekran tasarımında birçok nesne kullanımı gerçekleştirebilmektedir. Her nesnenin kendine özgü ayarları yapılmaktadır. Fazla detay istenilen projelerde özellikle ayar yapılmak istenilen nesnenin tasarım ekranından bulunması karmaşık bir hal alabilmektedir. Bu karmaşıklığı önlemek için tasarımda kullanılan bütün nesnelerin listesinin bulunduğu alandır. Bu sayede istenilen nesne seçilip Öznelik alanında ayarları gerçekleştirilebilmektedir.

AİRHMI LCD EKРАН EDITOR KILAVUZU

4.9 3.7.2 Nesnelerin Öznitelikleri Gösterim / Ayar Alanı

AirHMI Editör’de nesneler projeye dahil edildiklerinde otomatik olarak ilk ayarları ile eklenmektedir. Kullanıcılar kullanım amaçları ve istekleri doğrultusunda ekledikleri nesnelerin isimleri, boyutları, görünümleri, renkleri gibi birçok özelliğini bu alanda düzenleyebilmektedir.

4.10 ÖZİNİTELİKLERİN AÇIKLAMA ALANI



Nesnelerin ayarları öznitelik alanında gerçekleştirilmektedir. Fakat orada sadece öznitelik ismi yazmaktadır. Özniteliklerin Açıklama Alanında ise özniteliklerin açıklama kısmı bulunmaktadır. Öznitelik başlıklarının hangi işlevleri yerine getirdiği genel olarak açıklanmıştır.

4.11 KULLANICI PROJE KODU MENÜ ve ARAÇ ÇUBUKLAR



Tasarlanan projede en önemli kısım kod aşamasıdır. Proje temeline göre tasarım ekranında hangi durumlarda nelerin gösterileceği kodlama yapısı ile ayarlanmaktadır. Kod Menüsü kullanıcıya kod yazımında kodu kaydet, kopyala yapıştır, kod içerisinde anahtar kelime ara ve benzeri konularda yardımcı olabilecek bazı temel bileşenleri içermektedir.

4.12 KULLANICI PROJE KOD ALANI



AİRHMI LCD EKРАН EDITOR KILAVUZU

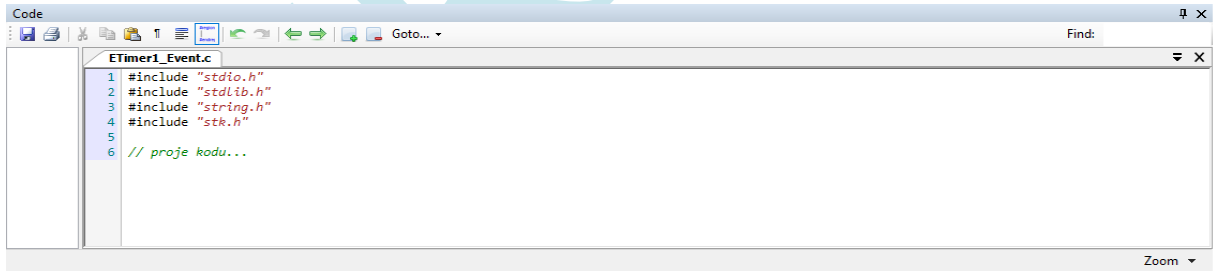
Kullanıcı Proje Kodu için geçerli bir AirHMI PICOC Kod Talimatını içerir. Bu bölüm programlamayı öğretmeyecek, ancak kullanıcının kod ekleyebilmesi için genel olarak yardımcı olacaktır. Bu alan içerisinde kullanıcılar ister Timer componentinin event'larına isterlerse de ekranda kullandıkları nesnelere event'larına C tabanlı kodları yazabileceklerdir. Screen Editor'un desteklediği hazır kütüphane kodları sayesinde yazılım zorluğu minimum seviyeye indirilen bu bölüm için hazır fonksiyonları üçüncü başlık altında (3. Fonksiyonlar) detaylı bir şekilde inceleyebilirsiniz. Orada belirtilen fonksiyonlara ek olarak C tabanlı kodların tamamı bu alana yazılarak programda eş zamanlı olarak çalıştırılabilmektedir.

4.13 KOD ALANI ZOOM ALANI



Proje tasarımında kod alanı yazı boyutunun kullanıcıya kullanımda kolaylık sağlaması için istenilen ölçüde yakınlaştırma ve uzaklaştırma yapabileceği alandır.

4.14 KOD ALANI



AirHMI Editör'ün çözüm odaklı, zaman ve efor konularında en verimli noktada tasarım oluşturmayı hedefleyen yapısının yanında en önemli avantajlarından biri de kolay ve anlaşılabilir kod yapısıdır. Kod yapısı C programlama dilinde hazırlanmıştır. Fakat kullanıcı odaklı olması ve kullanıcıya kullanımda kolaylık sağlayabilmesi için gerekli fonksiyonlar "stk.h" kütüphanesi altında hazırlanmıştır. Temel C kütüphanelerinin ekli olduğu bu düzende C programlama dilini kullanarak kodunuzu oluşturabilir ve gerekli fonksiyonları kodunuzun

AİRHMI LCD EKРАН EDITOR KILAVUZU

başına ekleyebilirsiniz. Hazır C fonksiyonlarına ek olarak nesnelere kontrol/ayar fonksiyonları, LCD ekran uyku modu, zamanlayıcı kod düzeni gibi önemli birçok konuda hazır fonksiyonları açıklamaları ile birlikte bu kılavuzda bulabilirsiniz. Burada önemli olan nokta bu fonksiyonların aktif olarak çalışabilmesi için “stk.h” kütüphanesinin her kod yapısının başına eklenmesi gerektiğidir.

```
ETimer1_Event.c
1 #include "stdio.h"
2 #include "stk.h"
3
4 char uartData[10];
5 int uartsiz;
6 uartDataGet(uartData, &uartsiz);
7
8 if(uartsiz > 0)
9 {
10     ImageSet ("EImage1" , "Visible" , "1");
11     LabelSet ("ELabel1" , "Caption" , "Deneme");
12     LocalIntVarSet("Variable1" , 2);
13
14     DrawScreenGet();
15
16 }
```

Örnek kod yapısı timer ile hazırlanmıştır. Timer kod yapısı için detaylı anlatım **2.1 TIMER** başlığı altında anlatılmaktadır.

Kod yapısı istenilen duruma göre Timer içerisinde olabileceği gibi Rezistif ekranlar için nesnelere dokunulduğunda çalışmasını istediğimiz kod yapısı da oluşturulabilmektedir. Timer içerisinde Event içerisinde oluşturacağınız kod zamanlayıcı aralığınıza tüm programda aktif olarak çalışırken nesnelere dokunulduğunda aktif olmasını istediğiniz kod yapısını aynı şekilde öznitelik kısmında bulunan OnUp kısmına eklenmesi gerekmektedir.

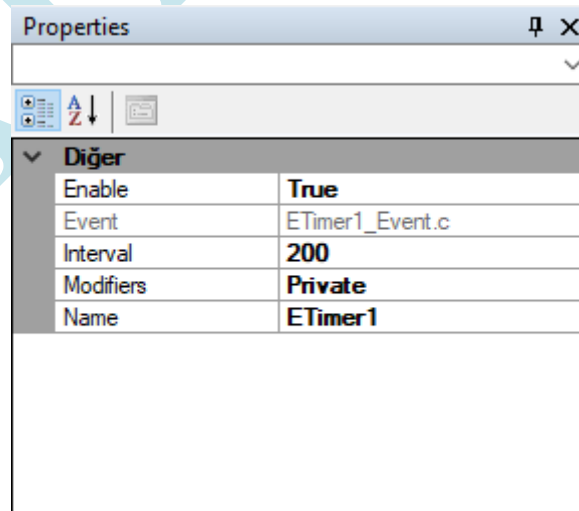
5. AİRHMI NESNELERİ VE FONKSİYONLAR

5.1 TIMER

Kod yapısı içerisinde belki de en önemli nokta Timer kullanımınıdır. Tasarlanan editör ekranının projede gerçek zamanlı çalışmasında oluşacak değişiklikler ve bu değişikliklerin hangi aralıklar ile olacağı Timer Özniteliklerinin içerisinde ayarlanmaktadır. Enable, Timer'ın aktif olup olmayacağını seçmektedir. Interval, milisaniye cinsinden hangi aralıklar ile kodun aktif olacağını seçildiği yerdir. Name, adında da anlaşılacağı gibi Timer'ın ismidir. Event bölümü ise proje tasarımı için oluşturulacak kod kısmını açma bölümüdür. ETimer1_Event.c ise oluşturulan kodun kaydedildiği C dosyasının ismidir.

Timer kullanımında kod yapısı, nesnelerin durumlarından bağımsız olarak Interval içerisinde ayarlanan süreye göre o aralıklarla kod dizinini aktif etmektedir. Kullanıcı eğer projesinde Rezistif bir ekran kullanıyor ve bir nesneye dokunulduğunda işlem yapmak istiyorsa; Dokunulduğunda işlem yapılmasını istediği nesnenin Öznitelikleri ayarlama kısmından OnUp kısmına gelip kodunu bu öznitelik altına eklemesi gerekmektedir. Böylece Timer'dan bağımsız olarak sadece o nesneye dokunulduğunda yazılan kod aktif olacaktır.

Timer Properties Penceresi



Dğer	
Enable	True
Event	ETimer1_Event.c
Interval	200
Modifiers	Private
Name	ETimer1

AİRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Enable	True False	Timer nesnesine enable yapar. Timer nesnesine disable yapar.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Event		Timer nesnesi yazılım alanıdır.
İnterval		Timer tekrar süresini ayarlar.
Modifiers	Private Public	Sadece bu sayfada çalışan timerdir. Tüm sayfalarda çalışan timerdir.

Fonksiyonlar

1. TimerSet ()

Açıklama

Buton nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

```
void TimerSet(unsigned char *name , unsigned char *type , unsigned char *value)
```

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

AİRHMI LCD EKLAN EDITOR KILAVUZU

Enable komutu

TimerSet(Nesne adı , “Enable” , “1 , 0 veya True , False”);

Örnek Kod:

```
ButtonSet ("Timer1" , "Enable" , "True");
```

Interval komutu

TimerSet(Nesne adı , “Interval” , “Milisaniye cinsinden deęer.”);

Örnek Kod:

```
ButtonSet ("Timer1" , "Interval" , "1000"); // interval 1 saniye olarak ayarlar.
```

AİRHMI

AIRHMI LCD EKLAN EDITOR KILAVUZU

5.2 Buton

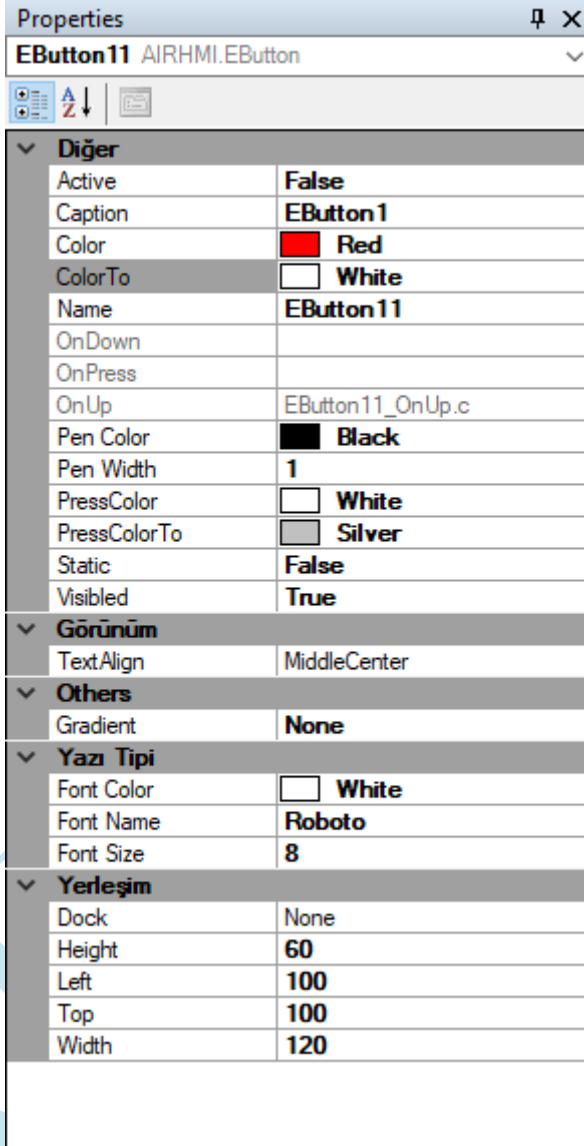
Buton nesnesi basıldıđı zaman herhangi bir iřlem yaptırmanı sađlayan nesnedir. Örneđin kullanıcıdan alınan veriyi bir yere göndermek, alınan veriyle iřlem yapmak veya mesaj verdirmek amacıyla kullanılabilir. Butonun konumunu istediđiniz yere sürükleyebilir ve boyutunu kenarlarından çekerek ayarlayabilirsiniz.

Buton Şekilleri



AIRHMI LCD EKLAN EDITOR KILAVUZU

Button Properties Penceresi



The screenshot shows a 'Properties' window for an 'EButton11' object. The window is titled 'Properties' and has a dropdown menu showing 'EButton11 AIRHMI.EButton'. Below the title bar, there are several icons for zooming and refreshing. The main area is a table of properties, organized into sections with expandable headers.

Diğer	
Active	False
Caption	EButton 1
Color	■ Red
ColorTo	■ White
Name	EButton11
OnDown	
OnPress	
OnUp	EButton11_OnUp.c
Pen Color	■ Black
Pen Width	1
PressColor	■ White
PressColorTo	■ Silver
Static	False
Visibled	True
Görünüm	
TextAlign	MiddleCenter
Others	
Gradient	None
Yazı Tipi	
Font Color	■ White
Font Name	Roboto
Font Size	8
Yerleşim	
Dock	None
Height	60
Left	100
Top	100
Width	120

AIRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Active	True False	Buton nesnesine basma işlevine izin verir. Buton nesnesi basma işlevine izin vermez.
Caption,Text		Buton nesnesinin ekranda gözüken adıdır.
Color		Buton nesnesinin ekrandaki rengini belirtir.
ColorTo		Gradient özelliği seçili olur ise, ekranda geçişli bir buton nesnesi oluşur. Bu nesnenin Color dan ColorTo ya geçiş rengini tanımlamak için kullanılır.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
OnDown		Buton nesnesine basma işlevi sırasında çalışan kod parçası buraya yazılır.
OnPress		Buton nesnesine elimizi basılı tuttuğumuz süreçte çalışacak olan kod parçasıdır. Tekrarlı olarak çalışır.
OnUp		Buton nesnesinden elimizi çekme anında çalışan kod parçası buraya yazılır.
Border Color		Buton Nesnesinin etrafının çizgi şeklinde sınırlarını belirtme rengidir.
Border Color		Buton nesnesinin etrafında oluşturulan çizginin kalınlığıdır.
Press Color		Buton nesnesinin basılı durumdaki ekrandaki rengini belirtir.
Press ColorTo		Gradient özelliği seçili olur ise, basılı durumda iken, ekranda geçişli bir buton nesnesi oluşur. Bu nesnenin Press Color dan Press ColorTo ya geçiş rengini tanımlamak için kullanılır.
Static		
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Text Aling		Buton nesnesi üzerindeki yazının butona göre konumlandırılmasıdır.
Gradient	None Top to Buttom Left to Right	Gradient özelliği kapalı olur. ColorTo ve Press ColorTo özelliği devre dışıdır. Gradient renkleri yukarıdan aşağı şeklinde uygulanır. Gradient renkleri soldan sağa doğru uygulanır.
Font Color		Butonun yazı rengidir.
Font Name		Buton nesnesi için farklı font seçenekleri tanımlama yapılır.
Font Size		Nesnenin yazısının fontunun büyüklüğüdür.
Dock		Buton nesnesinin ekrana yaslama şeklidir. Tam ekran şeklinde döşeme işlemi yapabilirsiniz.
Height		Nesnenin yüksekliğidir.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı
Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı

AİRHMI LCD EKLAN EDITOR KILAVUZU

Width	Nesnenin genişliğidir.
-------	------------------------

Fonksiyonlar

2. ButtonSet ()

Açıklama

Buton nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

void ButtonSet(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Visible ayarlama komutu

ButtonSet(Nesne adı , “Visible” , “1 , 0 veya True , False”);

Value özelliği “True” ayarlandığı zaman buton nesnesi gözüktür, “False” ayarlandığı zaman ise gözükmez.

Örnek Kod:

```
ButtonSet ("EButton1" , "Visible" , "True");
```

Active ayarlama komutu

ButtonSet(Nesne adı , “Active” , “1 , 0 veya True , False”);

AİRHMI LCD EKРАН EDITOR KILAVUZU

Örnek Kod:

```
ButtonSet("EButton1" , "Active" , "True");
```

Left ayarlama komutu

```
ButtonSet( Nesne adı , "Left" , "X koordinatı" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "Left" , "10");
```

Top ayarlama komutu

```
ButtonSet( Nesne adı , "Top" , "Y koordinatı" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "Top" , "255");
```

Width ayarlama komutu

```
ButtonSet( Nesne adı , "Width" , "Size ( 0 dan Ekran X boyutu kadar" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "Width" , "90");
```

Height ayarlama komutu

```
ButtonSet( Nesne adı , "Height" , "Size ( 0 dan Ekran Y boyutu kadar" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "Height" , "70");
```

Color ayarlama komutu

```
ButtonSet( Nesne adı , "Color" , "RGB Color hex formatında #RRGGBB" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "Color" , "#FFA07A");
```

ColorTo ayarlama komutu

```
ButtonSet( Nesne adı , "Color To" , "RGB Color hex formatında #RRGGBB" );
```

Örnek Kod:

```
ButtonSet("EButton1" , "ColorTo" , "#FFA07A");
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

Press_Color ayarlama komutu

ButtonSet(Nesne adı , “Press Color” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

```
ButtonSet("EButton1" , "Press_Color" , "#FFA07A");
```

Press_ColorTo ayarlama komutu

ButtonSet(Nesne adı , “Press ColorTo” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

```
ButtonSet("EButton1" , "Press_ColorTo" , "#FFA07A");
```

FontSize ayarlama komutu

ButtonSet(Nesne adı , “FontSize” , “Font size olarak 8-102 arasında ayarlanır.”);

Örnek Kod:

```
ButtonSet("EButton1" , "FontSize" , "12");
```

Font_Color ayarlama komutu

ButtonSet(Nesne adı , “Font Color” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

```
ButtonSet("EButton1" , "Font_Color" , "#FFA07A");
```

Caption ayarlama komutu

Buton nesnesinin ekranda görünen string ifadesi bu komut ile değiştirilir.

ButtonSet(Nesne adı , “Caption ve Text” , “Hello World!”);

Örnek Kod:

```
ButtonSet("EButton1" , "Caption" , "Hello World!");
```

```
ButtonSet("EButton1" , "Text" , "Hello World!");
```

AIRHMI LCD EKTRAN EDITOR KILAVUZU

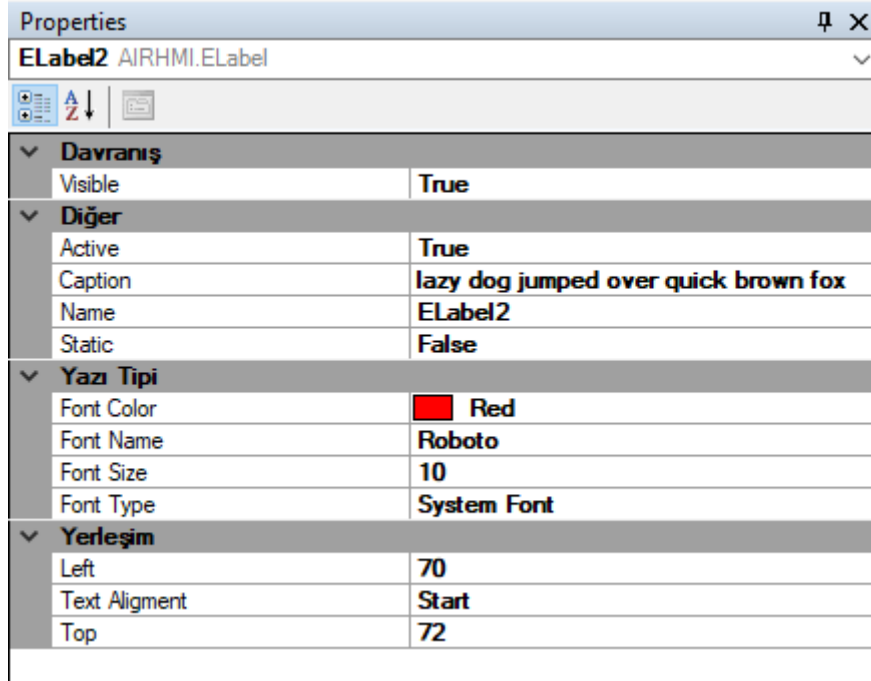
5.3 Label

Ekranı yazı yazma amacı ile kullanılan nesnedir. Font size olarak 8 den 102' ye kadar desteklemektedir. Default Font "Roboto" dur.



AIRHMI LCD EKРАН EDITOR KILAVUZU

Label Properties Penceresi



Özellik	Seçenek	Açıklama
Active	True False	Açık olması durumunda, label a dokunulduğu zaman klavye otomatik olarak çıkar. Klavye pasif durumdadır.
Caption ,Text		Label nesnesinin ekranda gözüken yazısıdır.
Color		Buton nesnesinin ekrandaki rengini belirtir.
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Static		Reserved.
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Text Alingment	Start Center	Label nesnesi sola dayama, Label nesnesi ortalama
Font Color		Labelin yazı rengidir.
Font Name		Label nesnesi için farklı font seçenekleri tanımlama yapılır.
Font Size		Nesnenin yazısının fontunun büyüklüğüdür.
Height		Nesnenin yüksekliğidir.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı

AİRHMI LCD EKРАН EDITOR KILAVUZU

Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı
Width		Nesnenin genişliğidir.

Fonksiyonlar

LabelSet ()

Açıklama

Label nesnesinin parametre ayarlarını düzenleyen komuttur.

void **LabelSet**(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Active ayarlama komutu

LabelSet(Nesne adı , “Active” , “1 , 0 veya True , False”);

Örnek Kod:

```
LabelSet("ELabell" , "Active" , "True");
```

Visible ayarlama komutu

LabelSet(Nesne adı , “Visible” , “1 , 0 veya True , False”);

Örnek Kod:

```
LabelSet("ELabell" , "Visible" , "1");
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

Left ayarlama komutu

LabelSet(Nesne adı , “Left” , “10”);

Örnek Kod:

LabelSet(*"ELabell"* , *"Left"* , *"10"*);

Top ayarlama komutu

LabelSet(Nesne adı , “Top” , “255”);

Örnek Kod:

LabelSet (*"ELabell"* , *"Top"* , *"255"*);

FontSize ayarlama komutu

LabelSet(Nesne adı , “FontSize” , “16”);

Örnek Kod:

LabelSet(*"ELabell"* , *"FontSize"* , *"16"*);

Font_Color ayarlama komutu

LabelSet (Nesne adı , “Font_Color” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

LabelSet(*"ELabell"* , *"Font_Color"* , *"#FFA07A"*);

Caption, Text ayarlama komutu

Label nesnesinin ekranda görünen string ifadesi bu komut ile değiştirilir.

LabelSet (Nesne adı , “Caption ve Text” , “Hello World!”);

LabelSet (*"ELabell"* , *"Caption"* , *"Hello World!"*);

LabelSet (*"ELabell"* , *"Text"* , *"Hello World!"*);

AİRHMI LCD EKCRAN EDITOR KILAVUZU

LabelGet()

void **LabelGet**(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değıştirilecek parametresinin ismi
value	Değıştirilecek parametrenin yeni alacağı değır

Caption, Text komutu

Label nesnesinin ekranda görünen string ifadesi bu komut ile değıştirilir.

```
LabelSet ( Nesne adı , "Caption ve Text" , char * buffer);  
Char value[20];  
LabelSet ("ELabel1" , "Caption" , value);  
LabelSet ("ELabel1" , "Text" , value);
```


AIRHMI LCD EKTRAN EDITOR KILAVUZU

5.4 Image

Image nesnesi resimleri gösterme ve resimleri buton olarak kullanma amacı ile kullanılabilir. Press image özelliği ile bir nesneye iki resim atayarak hiçbir kod yazmadan, normal durumda va press durumundaki resimlerini değiştirebilirsiniz.



AİRHMI LCD EKРАН EDITOR KILAVUZU

Image Properties Penceresi

Properties	
Elmage2 AIRHMI.Elmage	
Z ↓	
Davranış	
Visible	True
Diğer	
Active	True
Locked	False
Name	Elmage2
OnDown	
OnPress	
OnUp	Elmage2_OnUp.c
Opacity	100
PictureName	Asset 9.png
PicturePressImage	
ScaleX	0.5935
ScaleY	0.5984
Static	False
Yerleşim	
Dock	None
Height	73
Left	17
Top	242
Width	238

AIRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Active	True False	Açık olması durumunda resim buton gibi kullanılabilir. Kapalı olması durumunda sadece resim olarak kullanılır.ç
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Static		Reserved.
Locked	True False	Ekran a yerleştirilen nesnenin konumu değiştirmeye izin vermez. Resim istediğiniz konuma taşıyabilirsiniz.
Text Alingment	Start Center	Label nesnesi sola dayama, Label nesnesi ortalama
Height		Nesnenin yüksekliğidir.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı
Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı
Width		Nesnenin genişliğidir.
Image File		Bilgisayardan yüklemeniz gereken resim dosyasıdır.
Press Image File		Image nesnesine basılı tutarken ki resimdir.
ScaleX		Image nesnesin X boyutundaki büyütme ve küçültme oranıdır.
ScaleY		Image nesnesin Y boyutundaki büyütme ve küçültme oranıdır.
OnDown		Image nesnesine basma işlevi sırasında çalışan kod parçası buraya yazılır.
OnPress		Image nesnesine elimizi basılı tuttuğumuz sürece çalışacak olan kod parçasıdır. Tekrarlı olarak çalışır.
OnUp		Image nesnesinden elimizi çekme anında çalışan kod parçası buraya yazılır.

Fonksiyonlar

ImageSet ()

Açıklama

Image nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

```
void ImageSet(unsigned char *name , unsigned char *type , unsigned char *value)
```

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Örnek kod

Visible ayarlama komutu

```
ImageSet( Nesne adı , "Visible" , "1 , 0 veya True , False" );
```

Örnek Kod:

```
ImageSet("EImage1" , "Visible" , "True");
```

Left ayarlama komutu

```
ImageSet( Nesne adı , "Left" , "Left Pozisyonu" );
```

Örnek Kod:

```
ImageSet ("EImage1" , "Left" , "10");
```

AİRHMI LCD EKLAN EDITOR KILAVUZU

Top ayarlama komutu

ImageSet(Nesne adı , “Top” , “Top Pozisyonu”);

Örnek Kod:

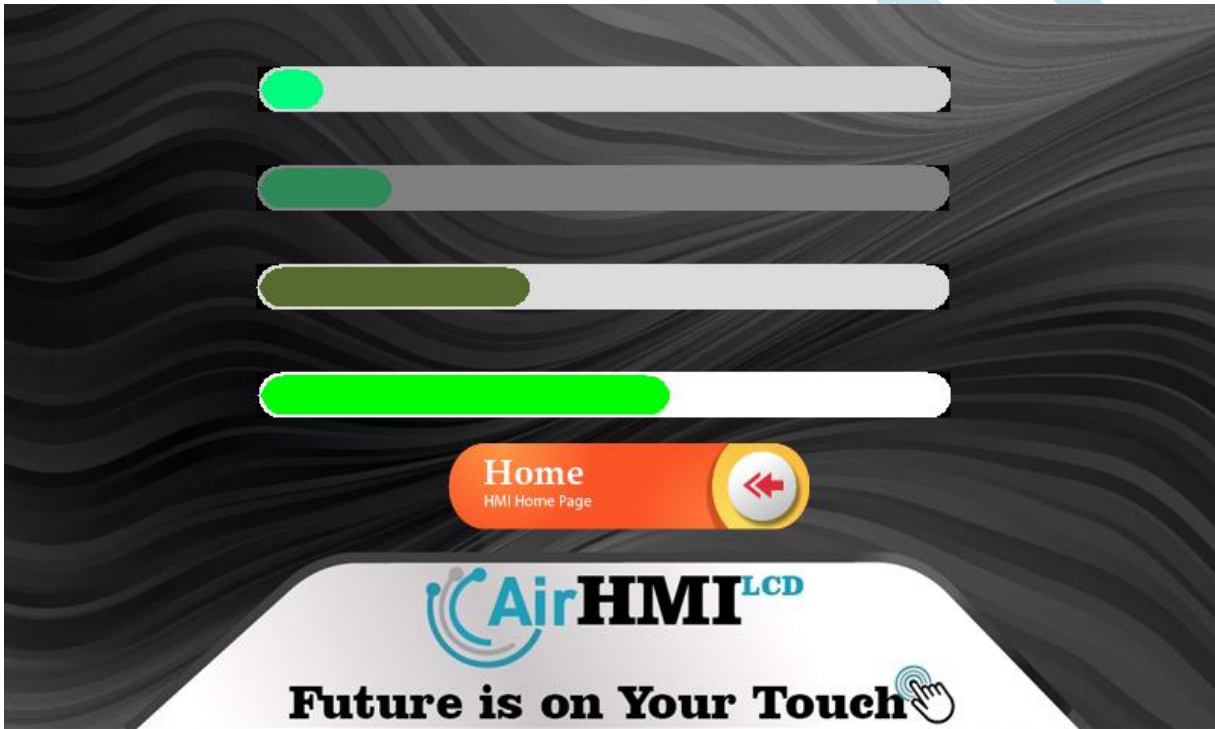
ImageSet (*"EImage1"* , *"Top"* , *"255"*);

AİRHMI

AIRHMI LCD EKTRAN EDITOR KILAVUZU

5.5 ProgressBar

Progress Bar ifadesi Türkçede “ilerleme çubuğu” anlamına gelmektedir. Uzun bir işlemin yürütülme aşamalarının grafiksel olarak gösterilmesi gerektiği durumlarda kullanılır. Progress Bar kullanımına örnek olarak: yürütülmekte olan bir video ya da ses dosyasının kalan zamanının Progress Bar üzerinde gösterilmesi, bir yakıt deposunun doluluk oranının Progress Bar kullanılarak grafiksel olarak gösterilmesi verilebilir.



AİRHMI LCD EKРАН EDITOR KILAVUZU

ProgressBar Properties Penceresi

Davranış	
Visible	True

Diğer	
BackgroundColor	<input type="text" value="White"/>
Color	<input type="text" value="Lime"/>
Flat	False
Name	ProgressBar1
Opacity	100
Range	100
Value	60

Yerleşim	
Height	30
Left	169
Top	244
Width	456

AİRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Color		Progressbar nesnesinin orta kısmında ilerleyen kısmın rengini belirtir.
BackgroundColor		Progressbar nesnesinin arka plan rengini belirtir.
Range		Progress bar toplam kaç değer olacağını belirtir.
Value		Progressbar in ilk ekrana yüklendiğinde yüzde kaçtan başlayacağını belirtir.
Height		Nesnenin yüksekliğidir.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı
Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı
Width		Nesnenin genişliğidir.

Fonksiyonlar

ProgressBarSet ()

Açıklama

Progress Bar nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

void ProgressBarSet(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Örnek kod

Visible ayarlama komutu

ProgressBarSet(Nesne adı , “Visible” , “1 , 0 veya True , False”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Visible" , "False");
```

Left ayarlama komutu

ProgressBarSet(Nesne adı , “Left” , “Ekrandaki X koordinatı pozisyonu”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Left" , "10");
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

Top ayarlama komutu

ProgressBarSet(Nesne adı , “Top” , “Ekrandaki Y koordinatı pozisyonu”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Top" , "255");
```

Color ayarlama komutu

ProgressBarSet(Nesne adı , “Color” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Color" , "255");
```

BackGround_Color ayarlama komutu

ProgressBarSet(Nesne adı , “BackGround_Color” , “RGB Color hex formatında #RRGGBB”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "BackGround_Color" , "1458269");
```

Range ayarlama komutu

ProgressBarSet(Nesne adı , “Range” , “Range (numeric)”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Range" , "100");
```

Value ayarlama komutu

ProgressBarSet(Nesne adı , “Value” , “Value (numeric)”);

Örnek Kod:

```
ProgressBarSet("ProgressBar1" , "Value" , "50");
```

AIRHMI LCD EKРАН EDITOR KILAVUZU

5.6 Slider

Kaydırıcı veya izleme çubuğu, kullanıcının bir göstereyi yatay veya dikey olarak hareket ettirerek bir değeri ayarlayabildiği grafiksel bir kontrol öğesidir. Bazı durumlarda, kullanıcı ayarı değiştirmek için kaydırıcıdaki bir noktaya da tıklayabilir.



AİRHMI LCD EKРАН EDITOR KILAVUZU

Slider Properties Penceresi

Properties	
Slider1 AIRHMI.EveSlider	
A Z ↓	
Davranış	
Visible	True
Diğer	
Active	True
BackgroundColor	DeepSkyBlue
Color	Gray
direction	vertical
Flat	False
Name	Slider1
OnDown	
OnUp	
Opacity	255
PressColor	128; 255; 128
Range	100
ThumbColor	Lavender
Value	50
Yerleşim	
Height	181
Left	196
Top	62
Width	56

AİRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Active	True False	Slider nesnesine basma işlevine izin verir. Slider nesnesi basma işlevine izin vermez.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Color		Slider nesnesinin arka kısmında kalan kısmının rengidir.
BackgroundColor		Slider nesnesinin arka plan rengini belirtir.
ThumpColor		Slider nesnesin üzerindeki yuvarlak kısmın rengidir.
PressColor		Slider nesnesine basıldığı zaman üzerindeki yuvarlak kısmın rengini değiştirir.
Range		Progress bar toplam kaç değer olacağını belirtir.
Value		Progressbar in ilk ekrana yüklendiğinde yüzde kaçtan başlayacağını belirtir.
Direction		Vertical , Horizontal Slider nesnesini ekranda kontrol yönünü belirtir.
Height		Nesnenin yüksekliğidir.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı
Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı
Width		Nesnenin genişliğidir.

AİRHMI LCD EKРАН EDITOR KILAVUZU

SliderSet ()

Açıklama

Slider nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

```
void SliderSet(unsigned char *name , unsigned char *type , unsigned char *value)
```

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Visible ayarlama komutu

```
SliderSet( Nesne adı , "Visible" , "1 , 0 veya True , False" );
```

Örnek Kod:

```
SliderSet("Slider1" , "Visible" , "1");
```

Left ayarlama komutu

```
SliderSet( Nesne adı , "Left" , "Ekrandaki X koordinatı pozisyonu" );
```

Örnek Kod:

```
SliderSet("Slider1" , "Left" , "10");
```

Top ayarlama komutu

```
SliderSet( Nesne adı , "Top" , "Ekrandaki Y koordinatı pozisyonu" );
```

Örnek Kod:

```
SliderSet("Slider1" , "Top" , "255");
```

AİRHMI LCD EKLAN EDITOR KILAVUZU

SliderGet ()

Açıklama

Slider nesnesinin parametre ayarlarını almaya yarayan komuttur.

Fonksiyon

void SliderGet(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değıştirilecek parametresinin ismi
value	Değıştirilecek parametrenin yeni alacağı değer

Value komutu

```
SliderGet( Nesne adı , "Value" , "char * buffer" );
```

Örnek Kod:

```
char buffer[20];
```

```
SliderGet("Slider1" , "Value" , buffer);
```

AIRHMI LCD EKTRAN EDITOR KILAVUZU

5.7 Gauge

Gauge nesnesi analog deęerleri gstermek iin etkili bir nesnedir. Aynı zamanda hız gstergesi olarak da kullanılır.



AİRHMI LCD EKРАН EDITOR KILAVUZU

Gauge Properties Penceresi

Davranış	
Visible	True
Diğer	
Active	True
Color	■ Blue
Flat	False
MajorCount	10
MinorCount	5
Name	Gauge 1
OnDown	
OnUp	
PenColor	■ Red
PressColor	■ White
Radius	94
Range	100
Tag	255
TicksVisible	False
Value	0
Yerleşim	
Left	59
Top	39

AİRHMI LCD EKРАН EDITOR KILAVUZU

Özellik	Seçenek	Açıklama
Visible	True False	Ekran ilk oluştuğu zaman görünür. Ekran ilk oluştuğu zaman görünmez.
Name		Nesnenin tasarım için kullanılan adıdır. Kod kısmındaki nesne adı bölümünde bu isim kullanılır.
Color		Gauge nesnesinin arka kısmında kalan kısmının rengidir.
BackgroundColor		Slider nesnesinin arka plan rengini belirtir.
PressColor		Slider nesnesine basıldığı zaman üzerindeki yuvarlak kısmın rengini değiştirir.
Range		Progress bar toplam kaç değer olacağını belirtir.
Value		Progressbar in ilk ekrana yüklendiğinde yüzde kaçtan başlayacağını belirtir.
Radius		Gauge nesnesinin çapını ayarlar.
TicksVisible		Gauge nesnesinin etrafındaki çizgileri açıp kapatır.
Left		Ekran üzerindeki pozisyonu belirtir. X koordinatı
Top		Ekran üzerindeki pozisyonu belirtir. Y koordinatı

Fonksiyonlar

GaugeSet ()

Açıklama

Gauge nesnesinin parametre ayarlarını düzenleyen komuttur.

Fonksiyon

void GaugeSet(unsigned char *name , unsigned char *type , unsigned char *value)

Parametre	Açıklama
name	Nesnenin ismi
type	Nesnenin değiştirilecek parametresinin ismi
value	Değiştirilecek parametrenin yeni alacağı değer

Örnek kod

Visible ayarlama komutu

```
GaugeSet( Nesne adı , "Visible" , "1 , 0 veya True , False" );
```

Örnek Kod:

```
GaugeSet( "Gauge1" , "Visible" , "1" );
```

Left ayarlama komutu

```
GaugeSet( Nesne adı , "Left" , "Ekrandaki X koordinatı pozisyonu" );
```

Örnek Kod:

```
GaugeSet( "Gauge1" , "Left" , "10" );
```

AİRHMI LCD EKCRAN EDITOR KILAVUZU

Top ayarlama komutu

GaugeSet(Nesne adı , "Top" , "Ekcrandaki Y koordinatı pozisyonu");

Örnek Kod:

```
GaugeSet("Gauge1" , "Top" , "255");
```

Color ayarlama komutu

GaugeSet(Nesne adı , "BackGround_Color" , "RGB Color hex formatında #RRGGBB");

Örnek Kod:

```
GaugeSet("Gauge1" , "Color" , "#ffaa02");
```

Value ayarlama komutu

GaugeSet(Nesne adı , "Value" , "Value (numeric)");

Örnek Kod:

```
GaugeSet("Gauge1" , "Value" , "100");
```

Range ayarlama komutu

GaugeSet(Nesne adı , "Range" , "Value (numeric)");

Örnek Kod:

```
GaugeSet("Gauge1" , "Range" , "30");
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

5.8 VARIABLE

Diğer	
Data	
Modifiers	Private
Name	EVariable1
Type	String

Değişkenler kod yapısı içerisinde değişkenlerin son değerlerinin veya kod içerisinde her düzenlemede değerinin kaybolmamasının istendiği durumlar için çok önemli bir rol almaktadırlar. Kod yapısı genel itibari ile Timer her aktif olduğunda veya Rezistif ekranlı projelerde dokunmanın aktif olduğu durumlarda derlenip yeniden çalıştığı için içerisinde oluşturulan normal değişkenler kendini sıfırlamaktadır. Bir önceki konumdan veya durumdan veriler kullanılmak istenildiğinde bu durum kullanıcı için büyük sorunlar teşkil etmektedir. Böyle bir sorunun yaşanmasını engellemek için devreye değişkenler girmektedir. Değişkenlerin ismi Öznitelikler bölümünden Name başlığı ile verilmektedir. Kullanılmak istenilen değişkenin tipi ise Type başlığı altından char ise String, sayısal değer ise Integer olarak seçilmelidir. Bir diğer özelliği olan Modifiers, Öznitelikler kısmından kullanmak istediğimiz değişkenin Private (yerel) ya da Public (global) olacağı seçilmeli. Yerel-global ayrımı birden fazla ekran tasarımı kullanılacak projelerde yapılmaktadır. Tek bir ekranda çalışma gerçekleştirilecek ise Private (yerel) değişken istenilen durumu gerçekleştirebilmektedir. Fakat birden fazla ekran kullanmak istenilen projelerde örneğin ikinci ekranda bulunan bir değer birinci ekrana geçildiğinde de kullanılmak istenilirse burada Public (Global) değişken kullanılmalıdır. Değişkenlerin kod yapısı içerisinde kullanımına dair açıklamalar aşağıda yer almaktadır.

```
GlobalStdVarGet("EVariable1" , "string");
```

1 2 3 4 5

Değişkenin:

1. Global veya Local
2. String veya Integer

AİRHMI LCD EKРАН EDITOR KILAVUZU

3. Değerinin Set ya da Get edileceğini
4. İsmi
5. Yeni değeri veya eski değerinin alınacağı değişken

durumlarına göre istenilen fonksiyon kullanılmalıdır.

```
int value;
```

```
LocalStdVarSet("EVariable1" , "string"); // Local olan String değişkeni Set etme
```

```
GlobalIntVarGet("EVariable2", &value); // Global olan Integer değişkeni Get etme
```

LocalStdVarGet ()

Açıklama

Local(yerel) string veri okuma komutudur.

Fonksiyon

```
void LocalStdVarGet(unsigned char *name , unsigned char *value)
```

Parametre	Açıklama
name	Yerel değişkenin ismi
value	Yerel değişkenin atanacağı string

Örnek kod

```
#include "stdio.h"
```

```
#include "stk.h"
```

```
char data[200];
```

```
LocalStdVarGet("EVariable1" , data); // Local olan String değişkeni Get etme
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

LocalStdVarSet ()

Açıklama

Local(yerel) string değer atama komutudur.

Fonksiyon

void LocalStdVarSet(unsigned char *name , unsigned char *value)

Parametre	Açıklama
name	Yerel değişkenin ismi
value	Yerel değişkenin alacağı string

Örnek kod

```
#include "stdio.h"
#include "stk.h"
LocalStdVarSet("EVariable1" , "string"); // Local olan String değişkeni Set etme
```

AİRHMI LCD EKLAN EDITOR KILAVUZU

LocalIntVarGet ()

Açıklama

Local(yerel) integer veri okuma komutudur.

Fonksiyon

```
void LocalIntVarGet(unsigned char *name , int *value)
```

Parametre	Açıklama
name	Yerel değişkenin ismi
value	Yerel değişkenin atanacağı integer

Örnek kod

```
#include "stdio.h"
#include "stk.h"
int value;
LocalIntVarGet("EVariable2", &value); // Local olan Integer değişkeni Get etme
```


AİRHMI LCD EKРАН EDITOR KILAVUZU

LocalIntVarSet ()

Açıklama

Local(yerel) integer değer atama komutudur.

Fonksiyon

void LocalIntVarSet(unsigned char *name , int value)

Parametre	Açıklama
name	Yerel değişkenin ismi
value	Yerel değişkenin alacağı integer

Örnek kod

```
#include "stdio.h"
#include "stk.h"
int value = 5;
LocalIntVarSet("EVariable2", value); // Local olan Integer değişkeni Set etme
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

GlobalStdVarGet ()

Açıklama

Global string veri okuma komutudur.

Fonksiyon

```
void GlobalStdVarGet(unsigned char *name , unsigned char *value)
```

Parametre	Açıklama
name	Global değişkenin ismi
value	Global değişkenin atanacağı string

Örnek kod

```
#include "stdio.h"
#include "stk.h"
GlobalStdVarGet("EVariable1" , "string"); // Global olan String değişkeni Get etme
```

GlobalStdVarSet ()

Açıklama

Global string değer atama komutudur.

Fonksiyon

```
void GlobalStdVarSet(unsigned char *name , unsigned char *value)
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

Parametre	Açıklama
name	Global değişkenin ismi
value	Global değişkenin alacağı string

Örnek kod

```
#include "stdio.h"
```

```
#include "stk.h"
```

```
GlobalStdVarSet("EVariable1" , "string"); // Global olan String değişkeni Set etme
```

GlobalIntVarGet ()

Açıklama

Global integer veri okuma komutudur.

Fonksiyon

```
void GlobalIntVarGet(unsigned char *name , int *value)
```

Parametre	Açıklama
name	Global değişkenin ismi
value	Global değişkenin atanacağı integer

AİRHMI LCD EKTRAN EDITOR KILAVUZU

Örnek kod

```
#include "stdio.h"

#include "stk.h"

int value = 5;

GlobalIntVarGet("EVariable2", &value); // Global olan Integer değişkeni Get etme
```

GlobalIntVarSet ()

Açıklama

Global integer değer atama komutudur.

Fonksiyon

```
void GlobalIntVarSet(unsigned char *name , int value)
```

Parametre	Açıklama
name	Global değişkenin ismi
value	Global değişkenin alacağı integer

Örnek kod

```
#include "stdio.h"

#include "stk.h"

int value = 10;

GlobalIntVarSet("EVariable2", value); // Global olan Integer değişkeni Set etme
```

AİRHMI LCD EKРАН EDITOR KILAVUZU

VariableSave()

Açıklama

Variable'i ekran içerisindeki hafızaya kayıt eder. Bu sayede ekran kapanıp açılrsa bile bu variable değeri kalıcı olarak hafızada tutulur. Variable içeriğinde değişiklik yaptıktan sonra tekrar kayıt etmek için aynı fonksiyon tekrar çağırılır. Maksimum 256 adet variable hafızaya kayıt edilebilir.

Fonksiyon

```
void VariableSave(unsigned char *name )
```

Parametre	Açıklama
name	değişkenin ismi

Örnek kod

```
#include "stdio.h"
#include "stk.h"
VariableSave("EVariable1"); //
```

5.9 Delay()

Açıklama

Kullanıldığı satırda belirlenen süre kadar beklemeyi sağlayan komuttur.

AIRHMI LCD EKTRAN EDITOR KILAVUZU

Fonksiyon

void Delay (int ms)

Parametre	Açıklama
ms	Zaman periyodunu belirtir

Örnek kod

```
#include "stk.h"  
Delay(1000);
```

5.10 uartDataGet ()

Açıklama

UART'tan gelen verilere göre AMHI Editör ekranında işlemler yapılabilmektedir. Kod düzeni içerisinde UART'tan gelen veriyi alma komutudur.

AİRHMI LCD EKРАН EDITOR KILAVUZU

Fonksiyon

void uartDataGet(char *value , int *uartsizе)

Parametre	Açıklama
value	UART'tan gelecek verinin depolanacağı string
uartsizе	UART'tan gelen verinin boyutu

Örnek kod

```
#include "stdio.h"

#include "stk.h"

char uartData[3000]; // Uarttan gelecek verinin depolanacağı
string

int uartsizе; // Uarttan gelen verinin boyutu

uartDataGet(uartData , &uartsizе); // Uarttan gelen verinin okunması
```

5.11 ChangeScreenSet ()

Açıklama

Kod içerisinde bulunan ekranlar arasında geçiş yapmayı sağlayan komuttur.

Fonksiyon

AİRHMI LCD EKРАН EDITOR KILAVUZU

void ChangeScreenSet(unsigned char *value)

Parametre	Açıklama
value	Geçiş yapılacak ekranın ismi

Örnek kod

```
#include "stk.h"
```

```
ChangeScreenSet("Screen1");
```

5.12 dateSet ()

Açıklama

RTC'de tarih verilerini yenileme/ayarlama komutudur.

Fonksiyon

void dateSet (unsigned char *days , unsigned char *months , unsigned char *years)

Parametre	Açıklama
days	Gün
months	Ay
years	Yıl

AİRHMI LCD EKLAN EDITOR KILAVUZU

Örnek kod

```
#include "stdio.h"

#include "stk.h"

unsigned char day, month, year;    // Kod dizininde örnek Tarih-Saat değişkenleri

day = 10;
month = 2;
year = 19;

dateSet(&day, &month , &year);    // RTC den Tarih verilerini ayarlama
```

5.13 timeSet ()

Açıklama

RTC'de saat verilerini yenileme/ayarlama komutudur.

Fonksiyon

```
void timeSet(unsigned char *hours , unsigned char *mins )
```

Parametre	Açıklama
hours	Saat
mins	Dakika

Örnek kod

```
#include "stdio.h"
#include "stk.h"
```

AİRHMI LCD EKTRAN EDITOR KILAVUZU

```
unsigned char hour, min;           // Kod dizininde örnek Tarih-Saat değişkenleri

hour = 16;
min = 30;

timeSet(&hour , &min);           // RTC de Saat verilerini yenileme/ayarlama
```

5.14 dateGet ()

Açıklama

RTC'den tarih verilerini alma komutudur.

Fonksiyon

```
void dateGet( unsigned char *days , unsigned char *months , unsigned char *years )
```

Parametre	Açıklama
days	Gün
months	Ay
years	Yıl

Örnek kod

```
#include "stdio.h"

#include "stk.h"

unsigned char day, month, year;     // Kod dizininde örnek Tarih-Saat değişkenleri

dateGet(&day, &month , &year);     // RTC den Tarih verilerini alma
```

AİRHMI LCD EKLAN EDITOR KILAVUZU

5.15 timeGet ()

Açıklama

RTC'den saat verilerini alma komutudur.

Fonksiyon

void timeGet(unsigned char *hours , unsigned char *mins)

Parametre	Açıklama
hours	Saat
mins	Dakika

Örnek kod

```
#include "stdio.h"
#include "stk.h"

unsigned char hour, min;           // Kod dizininde örnek Tarih-Saat değişkenleri
timeSet(&hour , &min);          // RTC de Saat verilerini okuma
```

5.16 AudioPlay()

Açıklama

AİRHMI LCD EKLAN EDITOR KILAVUZU

Kullanıcı, çalmak isteđi ses dosyasını AirHMI Editör üzerinden projeye ekledikten sonra bu fonksiyon ile çalma işlemini gerçekleştirebilmektedir.

Fonksiyon

void AudioPlay(unsigned char *audioname , unsigned char volume)

Parametre	Açıklama
audioname	Ses dosyasını ismi
volume	Ses düzeyi

Örnek kod

```
#include "stdio.h"
#include "stk.h"
int volume; // Ses Düzeyi
AudioPlay("SesDosyasınınİsmi" , volume );
```

5.17 AudioStop()

Açıklama

O anda çalınan ses işleminin sonlandırmak için kullanılır.

Fonksiyon

void AudioStop ();

Parametre	Açıklama
-----------	----------

AİRHMI LCD EKTRAN EDITOR KILAVUZU

Örnek kod

```
#include "stdio.h"
```

```
#include "stk.h"
```

```
AudioStop();
```

5.18 AudioStatusGet()

Açıklama

Ses dosyasının o anda çalınıp çalınmadığını ayarlar.

Fonksiyon

```
void AudioStatusGet(int *value)
```

Parametre	Açıklama
-----------	----------

AİRHMI LCD EKРАН EDITOR KILAVUZU

value	Player durumu (1 ses dosyası çalmaya devam ediyor , 0 ses dosyası çalma işlemi bitmiştir.
-------	---

Durum sorgulama komutu

AudioStatusGet(int *value);

Value özelliği “True” ayarlandığı zaman buton nesnesi gözükür, “False” ayarlandığı zaman ise gözükmez.

Örnek Kod:

```
int value;
```

```
AudioStatusGet(&value);
```

5.19 File_write ()

Açıklama

Flash’a yazma komutudur.

Fonksiyon

```
void File_write(unsigned char *name , void *buffer ,int size , int nmemb)
```

Parametre	Açıklama
name	Kullanılacak .txt dosyasının ismi

AİRHMI LCD EKLAN EDITOR KILAVUZU

buffer	String dizisinin ismi
size	Yazılacak dizinin boyutu
nmemb	1

Örnek kod

```
#include "stdio.h"
#include "stk.h"
char x_file[200];
memset(x_file , 0x00 , sizeof(x_file));
sprintf(x_file , "%s" , "Hello World !!!");

File_write("Message.txt" , x_file , sizeof(x_file), 1);

// Flashta Message.txt isimli bir dosya oluřturuldu ve bu dosya ierisine x_file
verisi sizeof(x_file) boyutu kadar yazıldı.
```

5.20 File_read()

Aıklama

Flash'tan okuma komutudur.

Fonksiyon

```
void File_read(unsigned char *name , void *buffer ,int size , int nmemb)
```

Parametre	Aıklama
name	Kullanılacak .txt dosyasının ismi
buffer	String dizisinin ismi

AİRHMI LCD EKРАН EDITOR KILAVUZU

size	Okuma boyutu
nmemb	1

Örnek kod

```
#include "stdio.h"
#include "stk.h"

char x_file[200];
memset(x_file , 0x00 , sizeof(x_file));

File_write("Message.txt" , x_file , sizeof(x_file), 1);

// Flashta bulunan Message.txt isimli bir dosyanın içerisinde ki verilerden
sizeof(x_file) kadarı x_file değişkenine okundu.
```

5.21 File_size()

Açıklama

Dosya boyutunu öğrenme komutudur.

Fonksiyon

```
void File_size(unsigned char *name ,int *size)
```

Parametre	Açıklama
name	Kullanılacak dosyanın ismi
size	Dosya boyutunun içinde tutulacağı integer bir değişken

Örnek kod

AİRHMI LCD EKLAN EDITOR KILAVUZU

```
#include "stdio.h"
#include "stk.h"

int f_size;

File_size("Message.txt" , &f_size); // Flashta bulunan Message.txt dosyasının
boyutunu öğrenme.
```

5.22 GPIO_Write()

Açıklama

Fonksiyon

```
void GPIO_Write(unsigned char *portName ,int value)
```

Parametre	Açıklama
portName	Gpio port
value	1 veya 0

Örnek kod

GPIO yazma komutu

```
GPIO_Write( GPIO adi , 1 veya 0 );
```

AİRHMI LCD EKLAN EDITOR KILAVUZU

Örnek Kod:

```
GPIO_Write( "GPIO_1" , 1 );  
GPIO_Write( "GPIO_1" , 0 );
```

5.23 GPIO_Read()

Açıklama

Fonksiyon

```
void GPIO_Read(unsigned char *portName ,int *value)
```

Parametre	Açıklama
portName	Gpio port
value	1 veya 0

Örnek kod

GPIO okuma komutu

```
GPIO_Read( GPIO adi , int * );
```

Örnek Kod:

```
int value;  
GPIO_Read( "GPIO_1" , &value );
```

AIRHMI LCD EKРАН EDITOR KILAVUZU

5.24 PWM_Set()

Açıklama

Airhmi ekran üzerinde 2 adet pwm çıkışı vardır. Bu fonksiyon ile pwm frekans duty ayarlanır.

Fonksiyon

```
void PWM_Set(int ch , int freq , int duty);
```

Parametre	Açıklama
ch	Pwm kanalı 1 veya 0
freq	Pwm frekansı
duty	Pwm 1 ,0 yüzdesidir. Değeri 0-100 olarak verilir.

Örnek kod

PWM komutu

```
PWM_Set(int ch , int freq , int duty);
```

Örnek Kod:

```
PWM_Set( 0,1000000, 50 ); // Channel 0 , 1Mhz %50 duty.  
PWM_Set( 1,2000000, 70 ); // Channel 0 , 2Mhz %70 duty.
```

AIRHMI LCD EKРАН EDITOR KILAVUZU

5.25 BuzzerSet()

Açıklama

Airhmi ekran dahili buzzer a sahiptir.

Fonksiyon

void BuzzerSet(int interval)

Parametre	Açıklama
interval	Milisaniye cinsinden buzzer çalma süresidir.

Örnek kod

Buzzer komutu

void BuzzerSet(int interval)

Örnek Kod:

BuzzerSet(100); // 100 ms buzzer set edilir.