Digispark, Arduino ailesinin en küçük üyelerinden biridir. Elektroniğe giriş için harika bir başlangıç, ya da diğer Arduinoların bir iş için çok büyük olduğu durumlarda muhteşem bir parçadır.

Digispark’ın şu özellikleri vardır:

* Arduino IDE 1.0+ (OSX/Win/Linux) için destek
* USB ile güç sağlama (5v veya 7-35v olmak üzere dış kaynak da olabilir, 12v veya daha azı tavsiye edilir)
* Eklenmiş 500ma 5V regülatör
* USB girişi
* 6 input/output pinleri (Eğer programınız USB aracılığıyla aktif biçimde iletişim kuruyorsa ikisi sadece USB tarafından kullanılmaktadır, yoksa USB aracılığıyla programlıyor olsanız bile tüm 6 pini de kullanabilirsiniz)
* 8k flash hafıza (bootloader’dan sonra yaklaşık 6k)
* I2C ve SPI (USI aracılığıyla)
* 3 pinde PWM (Pulse Width Modulation, Atım Genişlik Modülasyonu, Yazılım PWM ile daha mümkün)
* 4 pinde ADC (Analog to Digital Converter, Analog’tan dijitale dönüştürücü)
* Güç LED’i ve Test/durum LED’i
* Ayrıca Model A’da (elimizdeki model) pin 1’de, Model B’de ise pin 0’da çoktan bağlanmış küçük bir LED var digispark’ın üstünde.

Ayrıca digispark’ı kullanabilmek için önceden indirmiş olduğunuz Arduino programında yapmanız gereken birkaç işlem var.

Arduino programını açıp üst köşedeki “Dosya” üzerinden Tercihler’e tıklayın, “Ek devre kartları yöneticisi URL’leri” yazan yere şunu ekleyin (eğer çoktan eklenmiş linkler varsa yan taraftaki kutucuğa tıklayıp öyle ekleyin): [*http://digistump.com/package\_digistump\_index.json*](http://digistump.com/package_digistump_index.json)

Ardından tamam’a tıklayıp, yine üstteki seçeneklerden “Araçlar”a, ardından “Kart” seçeneği üzerine gelip “Kart Yöneticisi” seçeneğine tıklayın. Açılan pencerede sağ üstteki tip menüsünden “Contributed” seçeneğini seçin. “Digistump AVR Boards” paketini seçip indirin.

Bunlar tamamlandıktan sonra ise yine Araçlar seçeneğinden “Kart”tan “Digispark (Default - 16.5 mhz)” olan seçeneği seçin. Bu hangi Arduino üzerinde programlama yapacağınızı seçme işlemidir.

Artık Digispark’ınızda programlamaya hazırsınız! Sadece şunu hatırlayın, programınızı yüklemeden önce sağ üstteki tuşlardan “Kontrol Et”e tıklayın, ardından yüklerken “Yükle” tuşuna basın; alt panelde “Plug in your device now…” yazısını gördükten sonra Digispark’ınızı takın. Daha önce taksanız bir sorun yaşamazsınız, sadece programı yüklemez, daha geç takarsanız da time out hatası alabilirsiniz. Bu konuda bozmaktan korkmayın. Hadi başlayalım.

Öncelikle bu bağlanmış LED üzerinde bir şeyler yapalım.

# LED Blink

Arduino programını açtığınızda temel bir sketch penceresi açılır, ve burada nereye ne yazmanız gerektiği temel biçimde yazılı bulunmaktadır. Bu iki method dışında siz de kendi methodlarınızı oluşturabilirsiniz, ama bu şimdilik kalsın. Hatta bu deneme için minimum çaba sarf etmiş olacaksınız. Arduino penceresinin üst kısmından “Dosya”ya tıklayın, “Örnekler”den “01.Basics”e gelip “Blink”i seçin.

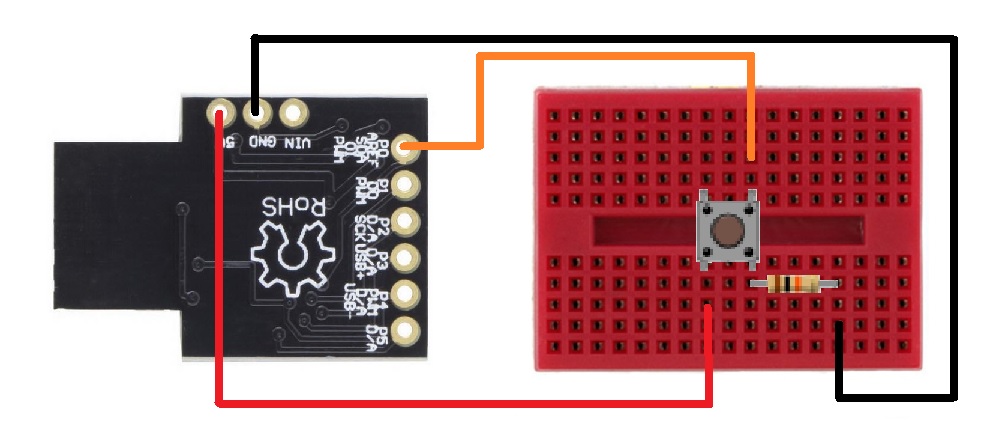
Yeni açılan pencerede hazır kod olacak. Burada önceden yazılmış olan pin 13. Siz her gördüğünüz 13’ün yerine sadece 1 yazacaksınız. “pinMode(1,OUTPUT);” ile 1. Pini output olarak kullanmak istediğinizi belirtmiş olacaksınız. “digitalWrite(1,HIGH);” ve “digitalWrite(1,LOW);” ile o pine akımı yüksek yapıp düşürüyorsunuz, “delay();” fonksiyonu da kodda gecikme, bekletme fonksiyonudur. Digispark milisaniye olarak girilen süre kadar bekler, o sırada hiçbir şey yapmaz. “1000” olarak yazılan 1 saniyedir, isterseniz değiştirebilirsiniz.

Artık ilk programınızı tamamladınız, “Yükle”ye tıklayıp alttaki panelde “Plug in your device now…” yazısını görünce Digispark’ınızı USB girişinize takabilirsiniz. Belki “can’t open device” şeklinde bir hata alabilirsiniz, bu durumda taktığınız port ile Arduino’nun arandığı port farklı olabilir. Bu durumda “Araçlar”dan “Port”a gelerek değiştirebilirsiniz. Programınızı yükleyin ve ilk denemenizi tamamlayın.

# Buton ile LED kontrolü

İlk denemenizi başarıyla tamamladınız mı? O zaman hadi bunu biraz daha geliştirelim. Bu defa başlangıç paketinizden 3 adet female-male (dişi-erkek) kablo ve butona ihtiyacınız olacak. Ayrıca bir adet 10k dirence daha ihtiyacınız olacak, akımı düşürebilmek için. Yoksa led’iniz kafasına göre yanıp sönmeye başlayabilir, çünkü butondan rastgele HIGH veya LOW değeri alacaktır. Bu küçük parçayı Robotik Kulübü’nden elde edebilirsiniz.

Bu çalışmanızda da yine hazır çalışmalardan birini kullanacağız, minimum değişikliklerle. Arduino penceresinden yine Dosya\Örnekler\02.Digital\Button çalışmasını seçin. Daha sonra da devrenizi tamamlayalım.



Yukarıdaki birleştirmeleri yaptıktan sonra, (Turuncu renkle gösterilmiş olan kabloyu Digispark’ta Vin, Gnd veya 5v dışındaki pinlere de bağlayabilirsiniz) açtığımız Button örneğini inceleyelim.

“const int” olarak tanımlanmış buttonPin ve ledPin’in değerlerini bizim çalışmamıza göre düzenlememiz gerek. Led olarak Digispark üzerindeki ledi kullanacağız, bunun için ledPin’in 1 olması gerekiyor. buttonPin değeri de yukarıdaki resimdeki turuncu renkle gösterilmiş kablonun ucunu hangi pine taktıysanız onun değerini yazmanız gerek. Resimdeki seçtiğimiz pin 0. Burada bu değerlerin “const int” olarak tanımlanmasının sebebi çalışmanın içinde asla değişmeyecek değerler, constant yani sabit değerler olmasıdır.

Örnek çalışmanın setup metodunda ise ledPin output olarak seçilmiş, ondan bir değer alınmayıp ona bir değer gönderildiği için; aynı şekilde de buttonPin input olarak seçilmiş.

Loop metoduna da “digitalRead()” fonksiyonu kullanılarak butondan 0 veya 1 (LOW veya HIGH) şeklinde değer alınması sağlanmış. Aslında burada loop metodunun içini daha kısa bir şekilde yazabilirsiniz. “digitalWrite(ledPin,digitalRead(buttonPin));” olarak değiştirebilirsiniz.

Şimdi bir çalışmanızı daha tamamladınız, modifiye ettiğimiz örneği Digispark’ınıza yükleyin, butona bastığınızda Digispark’ınızdaki led’in yanması gerekir.

Eğer isterseniz sahip olduğunuz ledlerinizle de bunu daha geliştirebilirsiniz, ancak o ledlerinizi kullanırken yine akım düşürmek için dirence ihtiyacınız olduğunu unutmayın.

# Bilgisayardan USB ile LED kontrolü

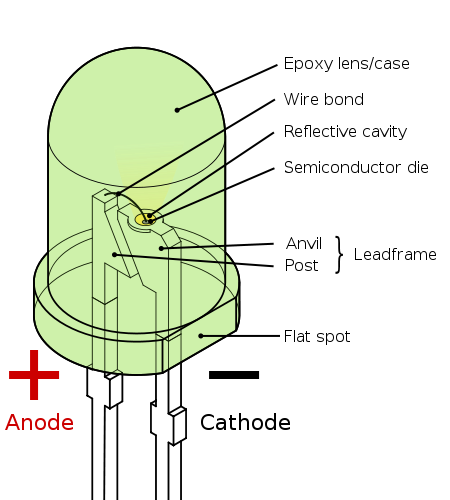
Eğer daha önce başka bir Arduino ile deneyiminiz varsa belki Arduino programının sağ üst köşesindeki Seri Port Ekranı aracılığıyla Arduino’nuzdan değer alıp değer gönderebileceğinizi biliyorsunuzdur. Ancak Digispark, Arduino ailesinin bu özelliğe sahip olmayan üyelerinden biridir. Bu nedenle Digispark’ınıza değer göndermek için farklı bir program kullanmanız gerekiyor. Şimdi öncelikle bunu kuralım.

Alttaki adresi kullanarak sağ üst köşeden “Fork” tuşuna tıkladıktan sonra açılan yeni sayfadan “clone or download” seçeneğinden indirebilirsiniz, ancak bunu yapabilmeniz için bir Github hesabına ihtiyacınız olacak. Github binlerce geliştiricinin bir araya gelip çalışmalarını kaydettiği bir platformdur, yani ileride de işinize yarayacaktır mutlaka.

https://github.com/digistump/DigisparkExamplePrograms/tree/master/Python/DigiUSB/windows

Eğer ZIP dosyası olarak indirdiyseniz bir dizine çıkartmanız gerekir. Bu aşamadan sonra, üstteki linkte gördüğünüz şekilde “DigisparkExamplePrograms/tree/master/Python/DigiUSB/windows” klasöründe “monitor.exe” adlı uygulamayı kullanacağız Digispark ile iletişim kurarken.

Bu çalışmada dilerseniz sadece kendi LEDleriniz ile bir devre oluşturabilir, veya biri için Digispark üzerindeki LED’i kullanabilirsiniz. Toplamda 2 adet LED kullanacağız.

LED için devre kurarken de dikkat etmeniz gerekenler şunlar: LED’inizin + ucuna bağlı bir direnç olmalı, direncin diğer ucu ise Digispark’taki kullanacağınız pin’e bağlı olmalı. LED’inizin + ucu da Digispark’taki GND pin’ine bağlı olmalı. + - kutuplarını buradan öğrenebilirsiniz (---->).

Programlama kısmını birlikte yapacağız. Öncelikle, Digispark ile iletişim kurmamızı sağlayacak olan Arduino kütüphanesi “DigiUSB” kütüphanesi. Programımızın başına “#include <DigiUSB.h>” eklememiz gerekiyor bu kütüphaneyi kullanabilmemiz için. Bu satırın hemen altında “int pin=0;” ekleyin, programlamanın sonraki aşamalarında kullanacağız.

Setup metodunun içine “DigiUSB.begin();”, “pinMode(0,OUTPUT);”, “pinMode(1,OUTPUT);” ekleyelim. Pinmode fonksiyonu daha önce öğrendiğiniz gibi pinlerinizi output veya input olarak ayarlamanıza yarıyor, DigiUSB.begin() ile de DigiUSB’yi başlatıyorsunuz.

Loop metodunda ise öncelikle DigiUSB’nin çalıştığını kontrol etmek için şu satırı ekleyelim: “DigiUSB.println(“Waiting for input…”);”. Daha sonra ise bu programda bilgisayardan input almak için şunları ekleyeyelim:

int lastRead;

while (true) {

if (DigiUSB.available()) {

lastRead = DigiUSB.read();

if (lastRead == '\n') {

break;

}

pin=lastRead;

}

DigiUSB.delay(100);

}

Şimdi, burada yaptığımız şey, bir döngü halinde eğer DigiUSB ile iletişim sağlanıyorsa ondan değer almak, bu değer yeni satır (yani enter) değilse bunu pin variable’ına kaydetmek. Eğer yeni satır ise döngüden çıkılması. Aslında bu sadece bir rakamı komut olarak almak için yeterli, ancak Digispark’ın hafızasından dolayı sadece bu kadar yapıyoruz. Ayrıca burada sadece pin sayısı olarak değer almak istiyoruz. Buradaki tek sorun DigiUSB değer alırken onu ASCII değeri olarak alıyor olması. Yani pin variable’ında girdiğiniz hane ASCII olarak tanımlı. Biz burada pin’i 0 ile 6 arasında olmasını istediğimiz için “pin=pin-48;” satırını bunların altına ekliyoruz.

Daha sonra ise LED açıp kapatma kısmı var. Bunu da ekledikten sonra programınız aynen böyle görünmeli:

#include <DigiUSB.h>

int pin = 0;

void setup() {

DigiUSB.begin();

pinMode(0,OUTPUT);

pinMode(1,OUTPUT); //Digispark üzerindeki LED’in pini

}

void loop() {

DigiUSB.println("Waiting for input...");

int lastRead;

while (true) {

if (DigiUSB.available()) {

lastRead = DigiUSB.read();

if (lastRead == '\n') {

break;

}

pin=lastRead;

}

DigiUSB.delay(100);

}

pin=pin-48;

analogWrite(pin,HIGH);

delay(1000);

analogWrite(pin,LOW);

delay(1000);

}

Şimdi bunu Digispark’ınıza yükleyin, ardından “monitor.exe”yi çalıştırın, ekranda “Waiting for input…” yazısını gördükten sonra 0 ve 1 göndererek LEDlerinizin yanıp sönmesini izleyin. Tebrikler! Bir çalışmayı daha tamamladınız!

# Mouse kontrolü

Evet, yanlış okumadınız, Digispark ile fare imlecini kontrol edebiliyorsunuz! Bir sonraki çalışmada da klavye kontrolünü öğreneceğiz, bu sayede arkadaşlarınıza küçük şakalar yapabilirsiniz.

Öncelikle, bunu yapmamızı sağlayan “DigiMouse.h” kütüphanesi var. Yapacağınız çalışmaların başına “#include <DigiMouse.h>” satırını eklemeyi unutmayın. Arduino Örnekler’inden “DigisparkMouse>Mouse” örneğini açtığınzda görebileceğiniz fonksiyonlar şunlar:

DigiMouse.moveX(); //parantezin içine -127 ile 127 arasında değer koyduğunuzda pozitif ise sağa, negatif ise sola hareket eder,

DigiMouse.moveY(); //aynı şekilde, pozitif ise aşağı, negatif ise yukarı hareket eder,

DigiMouse.scroll(); //aynı şekilde açık olan pencerede aşağı doğru kaydırır,

DigiMouse.move(X,Y,S); //ya da bu fonksiyonu kullanarak üstteki üçünü de yapabilirsiniz,

DigiMouse.leftClick(); //sol tıklama

DigiMouse.rightClick(); //sağ tıklama

DigiMouse.middleClick(); //orta tıklama

Ayrıca setButtons fonksiyonu da var, bu tıklama işlevlerinde kullanılması için; ancak anlaşılması biraz daha zor olduğu için kendi çalışmalarınızda şu metodları kullanın (size kolaylık için yazdık bunları, sadece setup metodundan sonraya eklemeniz yeterli, çağırmak içinse loop içinde sadece “clickerLeft();” şeklinde kullanmanız yeterli):

void clickerLeft(){

DigiMouse.setButtons(1<<0);

DigiMouse.delay(100);

DigiMouse.setButtons(0);

DigiMouse.leftClick();

DigiMouse.setButtons(0);

DigiMouse.delay(500);

}

void clickerRight(){

DigiMouse.setButtons(2<<0);

DigiMouse.delay(100);

DigiMouse.setButtons(0);

DigiMouse.rightClick();

DigiMouse.setButtons(0);

DigiMouse.delay(100);

}

void clickerMiddle(){

DigiMouse.setButtons(4<<0);

DigiMouse.delay(100);

DigiMouse.setButtons(0);

DigiMouse.middleClick();

DigiMouse.setButtons(0);

DigiMouse.delay(100);

}

Bu metodları ve diğer fonksiyonları kullanarak bilgisayarda istediğinizi yapabilirsiniz. Arkadaşlarınıza küçük şakaları yapabilirsiniz. Tabi ki setup metoduna “DigiMouse.begin();” satırını da eklemeyi unutmayın.

Şimdi de yapmış olduğumuz şu örneğin ne yaptığını tahmin etmeye çalışın, ardından Digispark’ınıza yükleyin ve deneyin. Bakalım tahmininiz doğru muymuş (Windows 10 işletim sisteminde kullanılması için ayarlanmıştır);

//#define \_\_AVR\_ATtiny85\_\_ eğer Arduino IDE 1.6.11 değilse

#include <DigiMouse.h>

void setup() {

DigiMouse.begin();

}

void clickerLeft() {

DigiMouse.setButtons(1<<0);

DigiMouse.delay(100);

DigiMouse.setButtons(0);

DigiMouse.leftClick();

DigiMouse.setButtons(0);

DigiMouse.delay(100);

}

void loop() {

for(int i=0;i<=15;i++){

DigiMouse.move(-100,50,0);

DigiMouse.delay(250);

}

clickerLeft();

DigiMouse.move(20,-40,0);

DigiMouse.delay(100);

clickerLeft();

DigiMouse.move(20,-40,0);

DigiMouse.delay(100);

clickerLeft();

}

# Keyboard kontrolü

Önceki çalışmanın sonundaki örneğimizi windows 10 bilgisayarınıza yükleyip denediyseniz kusurumuza bakmayın, biz de yapmadan edemedik.

Son çalışmamızda da klavye kontrolünü öğreneceğiz. Eğer Digispark’ınızdan ölçüm değerleri almak istiyorsanız bu DigiUSB kadar ideal. DigiUSB ile arasındaki fark olarak ise DigiUSB’den komutlar gönderebiliyorsunuz, DigiKeyboard ile sadece değerleri veya yazmasını istediğiniz şeyleri ekranınızda açık olan pencerenize yazıyor. Yani ihtiyacınız olan eğer açık olan pencerenize değerleri girmesiyse bunu kullanmanız daha faydalı. Tabi bir de yine küçük şakalar da hazırlayacaksanız…

Öncelikle bilmeniz gereken şeyler şunlar:

Eklemeniz gereken kütüphane “#include “DigiKeyboard.h””,

Setup metoduna herhangi bir şey eklemeniz gerekmiyor,

“DigiKeyboard.print();” veya “DigiKeyboard.println();” ile ekranda hangi pencere açıksa orada imlecin bulunduğu yere yazı yazdırabilirsiniz,

“DigiKeyboard.delay();” eklemeniz sayesinde bilgisayar klavyenin canlı ve bağlı olduğunu anlayabiliyor bu yüzden önemli,

“DigiKeyboard.sendKeyStroke(0);” fonksiyonu bazı eski sistemlerde bir delay’den sonra ilk harfin yazılmasının atlanmasını önlüyor, bu nedenle eğer doğru şekilde yazılmayacağını düşünüyorsanız eklemelisiniz.

Mutlaka dikkat edin ki ekranınızda açık olan pencereye yazılacak. Bu yüzden yazılmasını istediğiniz pencereyi açın.

Örnek olarak Arduino Örneklerdeki “DigiKeyboard>Keyboard” çalışmasını denemekle başlayabilirsiniz.