Açık Pencere Dedektörü

Zorluk: 🟡

Gerekli Malzemeler

- Manyetik sensörlü 1 adet **EMDACC4**
- 1 adet **mıknatıs** ve sabitlemek için biraz **bant**
- Her grup için 1 adet **Micro:bit**
- Her grup için 1 adet **Bilgisayar**

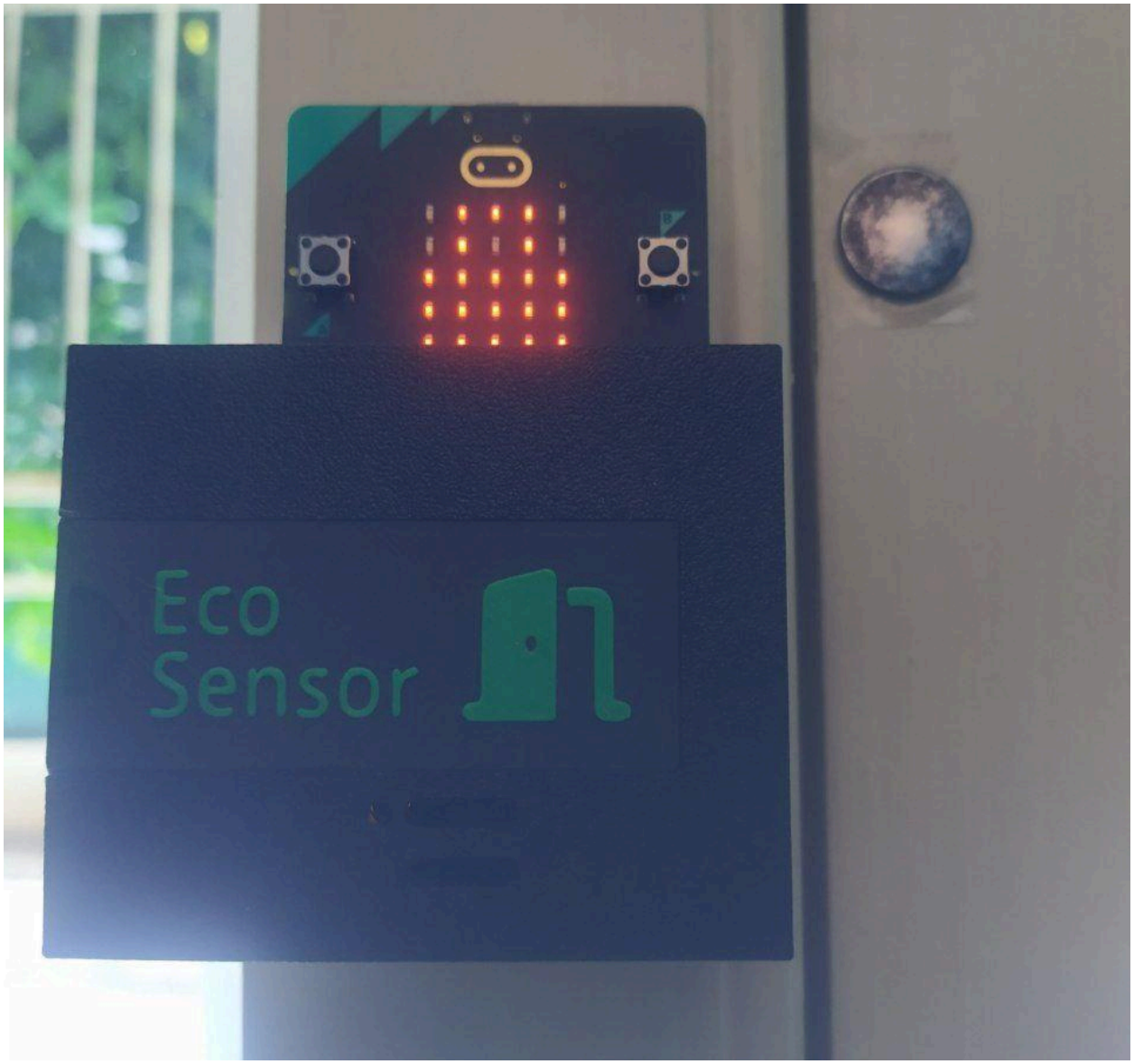
Ön Bilgiler

- **MakeCode** platformunda veri grafiği çizebilme becerisine sahip olmak
- Merkezi **EMDACC4**'e en son **firmware**'in yüklenmiş olması

1. Merkezi EMDACC4'ü Kurma

Merkezi cihaz olarak çalışacak Micro:bit'e firmware yüklendikten sonra, manyetik sensörle donatılmış **EMDACC4 modülünü** alın ve Micro:bit'i bu modüle bağlayın. Pili modüle takın ve kutuyu kapatın. Cihazı açın ve ekranda **kilit sembolü** görünüp görünmediğini kontrol edin. Bu sembolün görünmesi, kartın doğru şekilde bağlandığını ve manyetik sensör verilerini radyo dalgaları aracılığıyla yayınlamaya başladığını gösterir.

Şimdi EMDACC4 cihazını ve mıknatısı izlemek istediğiniz **kapı veya pencereye** sabitleyin. Aşağıdaki görseli referans olarak kullanabilirsiniz.



2. Micro:bit'i Bilgisayara Bağlama

Bilgisayarı açın ve bir web tarayıcısı başlatın (tercihen **Chrome**). https://makecode.microbit.org/ adresine gidin, yeni bir proje oluşturun ve **"Open Window Detector" (Pencere Açıklık Dedektörü)** gibi anlamlı bir isim verin.

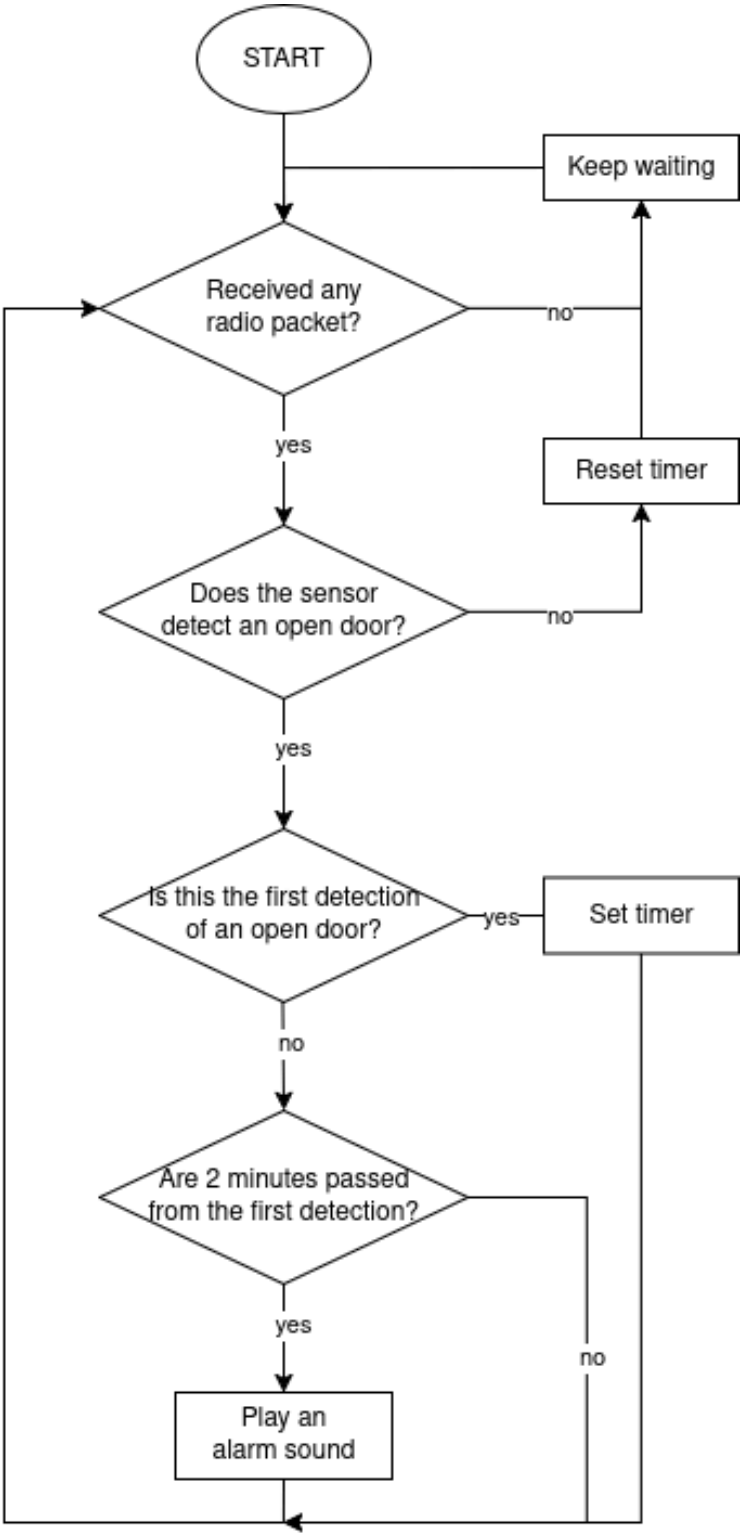
Micro:bit'i bilgisayarın **USB girişine** takın ve **MakeCode editörü** ile eşleştirin.

Sensörün yayınladığı değerleri görebilmek için basit bir **grafik çizim programı** yazın ve sensöre yakın bir mıknatısın varlığına göre bu değerlerin nasıl değiştiğini gözlemleyin.

3. Algoritmayı tasarlayın

Bu etkinliğin amacı, çevresel (peripheral) Micro:bit'in bir pencere ya da kapının çok uzun süre açık bırakıldığını algılaması ve böyle bir durumda bir uyarı göndermesidir. Bu, bir odayı soğutmak veya ısıtmak için kullanılan enerjinin israfını azaltmak adına kapı/pencereleri kapatma hatırlatıcısı olarak faydalı olabilir.

Nasıl çalışması gerektiğini hayal ederek başlayalım. Sürecin sonunda aşağıdaki akış şemasına benzer bir şey elde etmelisiniz.

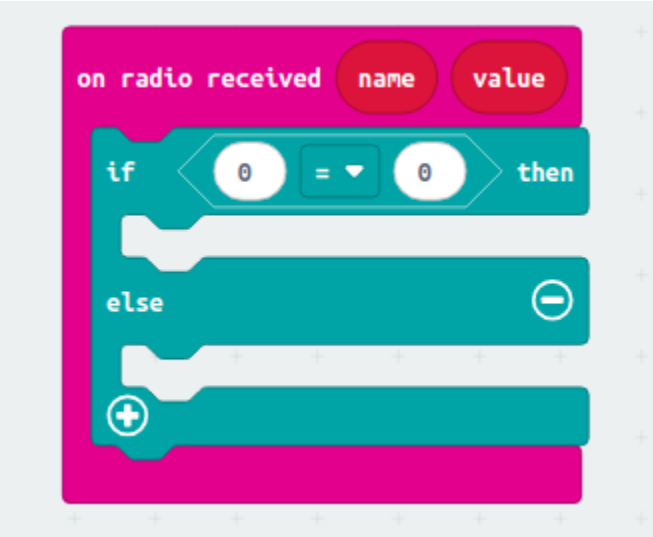


İlk denemede tam böyle olmazsa endişelenmeyin: algoritma yazma işi adım adım yapılır; önce belirsiz bir fikirle başlar, ardından ayrıntıları ekleyip köşe durumlarını kontrol ederek geliştirirsiniz. Koda acele etmeyin; önemli bir şeyi unutabilirsiniz!

4. Micro:bit'i Programlama

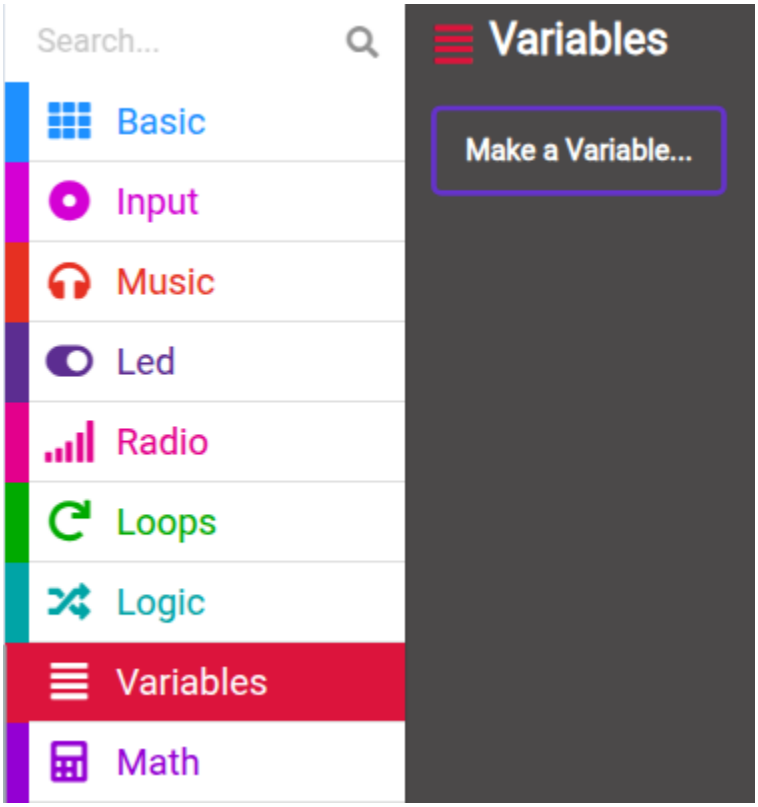
Artık algoritma netleştiğine göre, cihazın gerçek programlamasına geçebiliriz. Her şey, merkezi **EMDACC4** cihazından bir radyo paketi alınmasıyla başlar; bu nedenle **"on radio received <name> <value>"** bloğuna ihtiyacımız olacak.

Merkezi cihazdan her yeni paket aldığımızda bir koşulu değerlendirmemiz gerekir. Karar verme ve koşul değerlendirme ile ilgili tüm bloklar **"Logic" (Mantık)** menüsü altında bulunur. Özellikle, karşılaştırma koşulunu bağlayacağımız bir **"if … then … else" (eğer … ise … değilse)** bloğuna ihtiyacımız vardır.

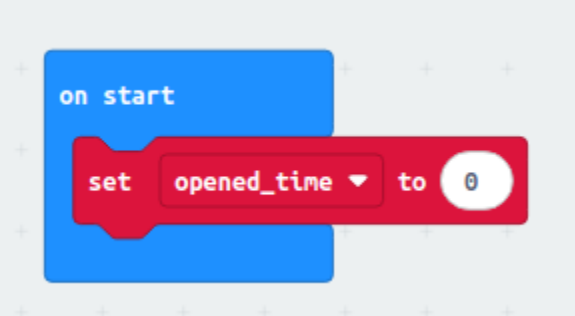


Değerlendirmemiz gereken koşul, kapının açık mı (değer **0**) yoksa kapalı mı (değer **1**) olduğunu belirlemektir.
Kapının durumunu ve bu durumdaki değişiklikleri takip edebilmek için, kapının **açıldığı zamanı kaydetmek** amacıyla kullanılacak bir değişken tanımlamamız gerekir.

Bu değişkeni **"Variables" (Değişkenler)** menüsünden **"Make a Variable" (Yeni Değişken Oluştur)** seçeneğine tıklayarak oluşturabilir ve adını **"opened_time"** olarak belirleyebilirsiniz.

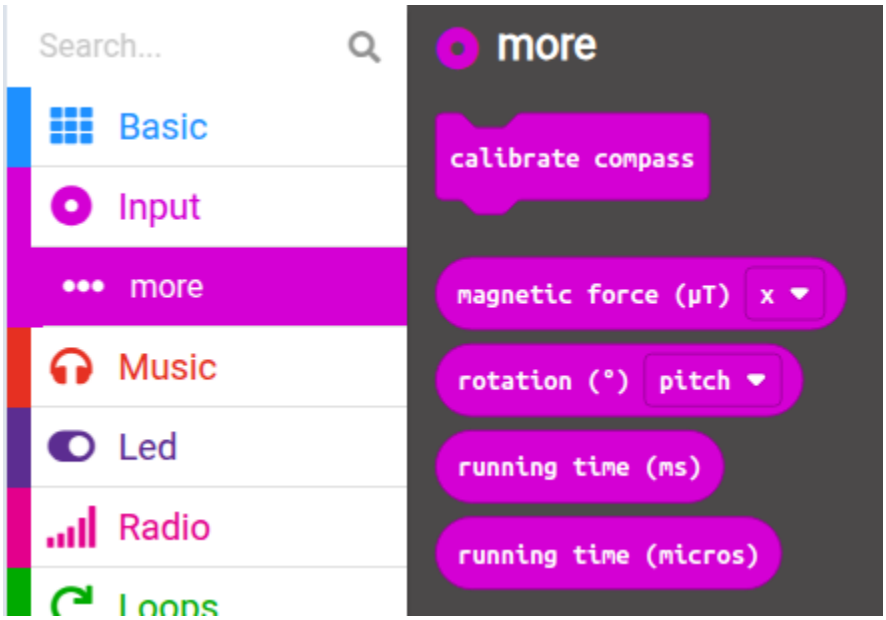


Her yeni değişken oluşturduğumuzda, ona bir başlangıç değeri atamamız gerekir ve bunu **“on start” (başlangıçta)** bloğu içinde yapmak genellikle iyi bir fikirdir.
Bu şekilde her şey düzenli ve kontrol edilebilir olur.



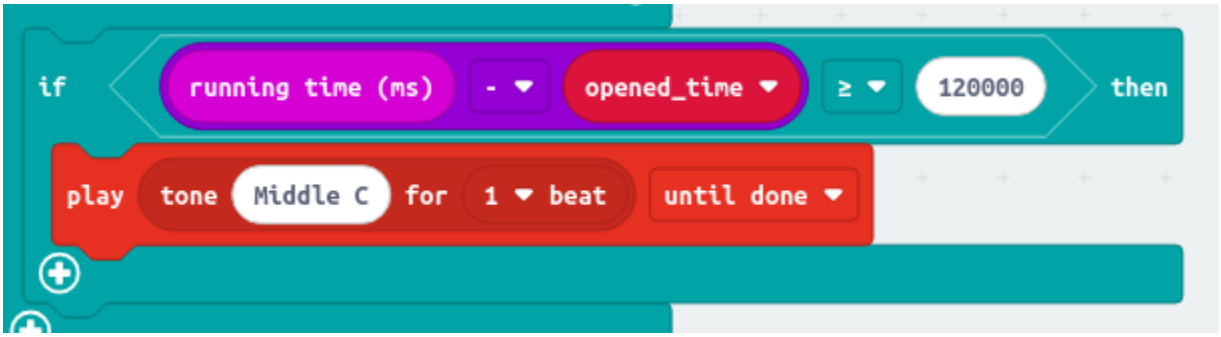
“opened_time” değişkenini **0** olarak ayarlamak, “kapı kapalı ve geçen zamanı takip etmeye gerek yok” demekle eşdeğerdir.
Bu komut, kapının kapalı olduğunu bildiren her paket alındığında tekrarlanacaktır.

Eğer kapının **açık** olduğunu tespit edersek ve daha önce **kapalı** olarak ölçülmüşse (yani başka bir koşul da devreye girer), **“opened_time”** değişkenini o anki **milisaniye (ms)** cinsinden zamana ayarlarız.
Bu değeri **“Input → more” (Girdi → daha fazla)** menüsünden alabiliriz.
Bu sayede kapının ilk kez ne zaman açıldığını kaydedebiliriz.



Akış şemasında açıklandığı şekilde kodu birleştirdiğimizde, kapının açık kaldığı süreyi ölçmemiz gereken bir noktaya geliriz.
Kapının açık olarak tespit edildiği zamanı zaten kaydettiğimiz için, geçen süreyi bulmak için **şimdiki zamandan bu değeri çıkarmamız** yeterlidir. Böylece kapının açık kaldığı **milisaniye cinsinden süreyi** elde ederiz.

Eğer kapı **2 dakikadan (120 saniye)** fazla açık kalırsa bir **alarm çalmak** istiyorsak, buna benzer bir kod yazmamız gerekir:



Koşul yerine getirildiğinde, **bir vuruşluk bir ses tonu** çalabiliriz; bu da **alarm** işlevi görecektir.
Kodu dikkatlice kontrol ettikten sonra, **Micro:bit'e yükleyip** beklendiği gibi çalışıp çalışmadığını test edebilirsiniz!

Fikirler

1. Uyarı tepkisini değiştirin: Ses çalmak yerine Micro:bit'in **yanıp sönmesini** veya **bir şey görüntülemesini** sağlayın.
2. Bunu başka verilerle birleştirin: Cihaz yalnızca **dış sıcaklık 18 °C'nin altındayken** pencerenin açık olduğunu bildirirse ne olurdu?

Referans Kodu

