

iDea

rsBASIC KOMUTLARI

İÇİNDEKİLER

2.1	Etiketler (Labels)	4
2.2	Yorumlar (Comments)	5
2.3	Semboller (Symbols)	6
2.4	backward (geri)	7
2.5	debug (hata ayıkla/izle)	8
2.6	dec (azalt)	9
2.7	do .. until	10
2.8	do .. while	11
2.9	for .. next	12
2.10	forward (ileri)	13
2.11	gosub (alt program)	14
2.12	halt (dur)	15
2.13	high (yüksek)	16
2.14	if. then (eğer..ise)	17
2.15	inc (artır)	18
2.16	let (ifade tanımla)	19
2.17	low (düşük)	20
2.18	motor	21
2.19	pause (bekle)	22
2.20	pulsout (atım gönder)	23
2.21	readadc (analog oku)	24

1 Genel Bakış

Bu dokümanda, robotsan tarafından geliştirilmiş olan *iDea* (integrated Development Environment for Applications – uygulamalar için bütünleşik geliştirme ortamı) geliştirme ortamı içerisinde yer alan rsBASIC dili komutları açıklanmaktadır.

Dokümanda komutlar Türkçe ve İngilizce karşılıkları parantez içinde verilerek anlatılmıştır. Örneğin: inc (artır) (inc, increment'in kısaltılmasıdır).

Bu doküman *iDea* yazılımı revizyon 2.1 kullanılarak hazırlanmıştır. Lütfen, www.robotsan.com.tr web adresinden güncellemeleri kontrol ediniz.

2 rsBASIC Komutları

2.1 Etiketler (Labels)

Etiketler program içerisinde belirli bölümleri işaret etmek ve *goto*, *gosub* gibi komutların bu işaretli noktalara gönderme yapmalarında kullanılırlar.

Etiketler küçük ve büyük harf barındırabilir, ancak Türkçe karakter içeremezler.

Örnek 1

dongu:	‘ Döngü başlangıç etiketi
high 1	‘ 1 numaralı çıkışı aç
pause 1000	‘ 1 sn bekle
low 1	‘ 1 numaralı çıkışı kapat
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

Örnek 2

AnaDongu:	‘ Ana döngü başlangıç etiketi
gosub MotorAc	‘ <i>MotorAc</i> etiketine git
pause 1000	‘ 1 sn bekle
gosub MotorKapat	‘ <i>MotorKapat</i> etiketine git
pause 2000	‘ 2 sn bekle
goto AnaDongu	‘ Başlangıca (AnaDongu etiketine) geri dön

MotorAc:	‘ Motor aç başlangıç etiketi
high 1	‘ 1 numaralı çıkışı aç
return	‘ Bu etiketi/döngüyü sonlandırıp 1 sn bekle komutuna döner

MotorKapat:	‘ Motor kapat başlangıç etiketi
low 1	‘ 1 numaralı çıkışı kapat
return	‘ Bu etiketi/döngüyü sonlandırıp 2 sn bekle komutuna döner

2.2 Yorumlar (Comments)

Yorumlar program içerisinde kullanıcı tarafından bilgilendirici açıklama yazabilmek amacıyla kullanılır ve derleyici tarafından dikkate alınmazlar.

rsBASIC derleyicisinde yorumlar kesme işareti (‘) ile başlar ve bulunduğu satırın sonuna kadar işlevliğini korur.

Yorumlar içerisinde Türkçe karakter kullanılabilir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
high 1	‘ 1 numaralı çıkışı aç
pause 1000	‘ 1 sn bekle
low 1	‘ 1 numaralı çıkışı kapat
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.3 Semboller (Symbols)

Semboller *iDea* yazılımında sabit değer ve değişken atamalarında kullanılırlar.

rsBASIC derleyicisinde sembol tanımlaması için *symbol* yazılır, isim verildikten sonrasında eşit işareti (=) ile atama yapılır.

Semboller program başlangıcında tanımlanır, tanımlama sırasında rakamlar ve alt çizgi kullanılabilir ancak Türkçe karakter içeremezler.

Sembol kullanımı program hafıza kullanımını artırmaz. *iDea* tarafından aşağıda listelenen tanımlar desteklenir:

bayt: b0..b15, **kelime:** w0..w15, **noktalı sayı:** f0..f15, **bacak:** pin2...pin7

Örnek 1

symbol MotorA = 2	‘ MotorA değişkeni 2 değeri ile sabit sayı olarak tanımlandı
symbol j = b0	‘ j değişkeni bayt (byte) olarak tanımlandı
symbol Sensor1 = w0	‘ Sensor1 değişkeni kelime (word) olarak tanımlandı
symbol Sensor2 = w1	‘ Sensor2 değişkeni kelime (word) olarak tanımlandı
symbol Sicaklik = f0	‘ Sicaklik değişkeni noktalı sayı (floating) olarak tanımlandı
symbol Sensor = pin3	‘ Sensor değişkeni <i>iDea kontrol kartı P6</i> girişi

Örnek 2

symbol Anahtar = pin2	‘ Anahtar değişkeni <i>iDea kontrol kartı P5</i> girişi
dongu:	‘ Döngü başlangıç etiketi
if Anahtar = 0 then	‘ if..then..else döngüsü (eğer Anahtar “0” ise)
high 1	‘ 1 numaralı çıkışı aç
else	‘ diğer durumda (eğer Anahtar “1” ise)
low 1	‘ 1 numaralı çıkışı kapat
endif	‘ end..if tanımlaması sona erdi
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.4 backward (geri)

a ve b olarak tanımlanmış *iDea kontrol kartı* motor çıkışlarını geri çevirmek için kullanılır. Motorun hızını belirlemek için sonrasında “motor” komutu kullanılmalıdır.

Kullanım Şekli

rsBASIC
backward motor

Açıklama

- “motor” a veya b olabilir.
- a motoru *iDea kontrol kartı* P1 ve P2 çıkışlarına bağlanmalıdır.
- b motoru *iDea kontrol kartı* P3 ve P4 çıkışlarına bağlanmalıdır.
- “backward a” komutu low P1, high P2 ile eşdeğerdir.
- “backward b” komutu low P3, high P4 ile eşdeğerdir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
backward a	‘ a motorunun yönünü geri olarak ayarla
motor a,90	‘ a motorunun hızını 90 olarak ayarla
pause 1000	‘ 1 sn bekle
forward a	‘ a motorunun yönünü ileri olarak ayarla
motor a,80	‘ a motorunun hızını 80 olarak ayarla
pause 1000	‘ 1 sn bekle
halt a	‘ a motorunu durdur
pause 1000	‘ 1 sn bekle
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.5 debug (hata ayıkla/izle)

iDea kontrol kartı analog girişlerinin değerlerini izlemek ve algoritma geliştirirken değişken değerlerini izleyerek hata ayıklamak için kullanılır.

Kullanım Şekli

rsBASIC
debug pause 100

Açıklama

- “debug” komutu iDea kontrol kartı üzerindeki analog portların ve 6 kelime tipindeki değişkenin değerini sırayla bilgisayara iletir.
- Verileri izlemek için PC ve iDea kontrol kartı USB kablosu ile bağlı olmalıdır. iDea kontrol kartı verileri kablo ile bilgisayarınıza iletilir.
- Veriler bilgisayarın yükleme yapılan COM portunda 9,600 baudrate hızı ile bilgisayara iletilir.
- “debug” komutunun ardından izlemeyi kolaylaştırmak için asgari 100ms bekleme konulmalıdır.
- “debug” komutu ile bilgisayara iletilen bilgiler aşağıdaki gibidir:
P11;P12;P13;P14;w0;w1;w2;w3;w4;w5
Pxx değerleri 0-1023 arasında analog/sayısal çevrim değerlerine karşılık gelecektir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
debug	‘ iDea kontrol kartı verilerini bilgisayara gönder
pause 500	‘ Gönderimler arasında 500 ms bekle
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.6 dec (azalt)

Sembol ile tanımlanmış bir değeri azaltmak için kullanılır.

Kullanım Şekli

rsBASIC
dec sembol

Örnek

symbol i = b0	‘ i değişkeni bayt olarak tanımlandı
i = 20	‘ i değişkenine 20 sabit değeri başlangıçta atandı
dongu:	‘ Döngü başlangıç etiketi
dec i	‘ i değişkenini bir artır
if i = 10 then	‘ if..then..else döngüsü (eğer i = 10 ise)
high 1	‘ 1 numaralı çıkışı aç
else	‘ diğer durumda (eğer i 10’dan farklı ise)
low 1	‘ 1 numaralı çıkışı kapat
endif	‘ end..if tanımlaması sona erdi
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.7 do .. until

Bir kod parçasının tekrar etmesinin istenildiği durumlarda kullanılır. (do until..loop) veya (do .. loop until) döngüsü içerisindeki kod parçası ifade geçerli olana kadar çalıştırılır.

Kullanım Şekli

rsBASIC
do until ifade
loop
do ifade
loop until

Açıklama

- “ifade” mantıksal bir karşılaştırmadır.
- “do..loop until”, “for..next” ve “do until”den farklı olarak ifadenin doğru olup olmadığını döngünün sonunda karşılaştırdığı için en az bir kere döngü içerisindeki kodu çalıştırır.

Örnek

```
symbol i = w0          ‘ i değişkenini kelime olarak tanımladık
i = 5                  ‘ i değişkenine 5 sabit değerini atadık
do until i < 2         ‘ i < 2 olana kadar loop’a kadarki kodu çalıştır
    high 1             ‘ 1 numaralı çıkışı aç
    dec i              ‘ i’nin değerini bir azalt
loop
do                    ‘ i > 2 olana kadar loop’a kadarki kodu çalıştır
    low 1              ‘ 1 numaralı çıkışı kapat
    inc i              ‘ i’nin değerini bir artır
loop until i > 2
```

v2.1

2.8 do .. while

Bir kod parçasının tekrar etmesinin istenildiği durumlarda kullanılır. (do while..loop) veya (do .. loop while) döngüsü içerisindeki kod parçası ifade geçersiz olana kadar çalıştırılır.

Kullanım Şekli

rsBASIC
do while ifade
loop
do ifade
loop while

Açıklama

- “ifade” mantıksal bir karşılaştırmadır.
- “do..loop while”, “for..next” ve “do while” dan farklı olarak ifadenin doğru olup olmadığını döngünün sonunda karşılaştırdığı için en az bir kere döngü içerisindeki kodu çalıştırır.

Örnek

```
symbol i = w0          ‘ i değişkenini kelime olarak tanımladık
i = 1                  ‘ i değişkenine 1 sabit değerini atadık
do while i < 2         ‘ i < 2 olduğu sürece loop’a kadarki kodu çalıştır
    high 1             ‘ 1 numaralı çıkışı aç
    inc i              ‘ i’nin değerini bir artır
loop
do                     ‘ i > 2 olduğu sürece loop’a kadarki kodu çalıştır
    low 1              ‘ 1 numaralı çıkışı kapat
    dec i             ‘ i’nin değerini bir azalt
loop while i > 2
```

v2.1

2.9 for .. next

Bir kod parçasının tekrar etmesinin istenildiği durumlarda kullanılır. for..next döngüsü içerisindeki kod parçası başlangıç-son değerleri arasında tekrarlayarak çalıştırılır.

Kullanım Şekli

rsBASIC
for değişken = başlangıç to son
next

Açıklama

- “başlangıç” değişkenin ilk değeri, “son” son değeridir.
- “değişken” istenilen son değere ulaşana kadar for..next arasında yer alan kod çalıştırılır. Her döngü çalıştırılışında “next” bölümüne gelindiğinde değişken bir artırılır.
- “değişken” bayt veya kelime olabilir.

Örnek

symbol i = b0	‘ i değişkenini bayt olarak tanımladık
dongu:	‘ Döngü başlangıç etiketi
for i = 1 to 10	‘ 10 kere 1 numaralı çıkışı açıp/kapatacağız
high 1	‘ 1 numaralı çıkışı aç
pause 500	‘ 500 ms bekle
low 1	‘ 1 numaralı çıkışı kapat
pause 500	‘500 ms bekle
next	‘ i’yi bir artır, gerekli koşul sağlanıyorsa for döngüsüne git
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.10 forward (ileri)

a ve b olarak tanımlanmış *iDea kontrol kartı* motor çıkışlarını ileri çevirmek için kullanılır. Motorun hızını belirlemek için sonrasında “motor” komutu kullanılmalıdır.

Kullanım Şekli

rsBASIC
forward motor

Açıklama

- “motor” a veya b olabilir.
- a motoru *iDea kontrol kartı* P1 ve P2 çıkışlarına bağlanmalıdır.
- b motoru *iDea kontrol kartı* P3 ve P4 çıkışlarına bağlanmalıdır.
- “forward a” komutu high P1, low P2 ile eşdeğerdir.
- “forward b” komutu high P3, low P4 ile eşdeğerdir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
backward a	‘ a motorunun yönünü geri olarak ayarla
motor a,90	‘ a motorunun hızını 90 olarak ayarla
pause 1000	‘ 1 sn bekle
forward a	‘ a motorunun yönünü ileri olarak ayarla
motor a,80	‘ a motorunun hızını 80 olarak ayarla
pause 1000	‘ 1 sn bekle
halt a	‘ a motorunu durdur
pause 1000	‘ 1 sn bekle
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.11 gosub (alt program)

Program içerisinde alt bir programa gönderme yapmak için kullanılır.

Kullanım Şekli

rsBASIC
gosub alt-program

Açıklama

- “alt-program” bir etiket ismi olmalıdır.
- Alt programdan geri dönmek için “return” komutu kullanılır.
- *gosub*, fonksiyonları gruplayarak tekrar eden program parçalarını alt bir programda birleştirmek için kullanılır.

Örnek

dongu: ‘ Döngü başlangıç etiketi

 gosub LEDYak ‘ LEDYak etiketli alt-programa git

 pause 1000 ‘ 1 sn bekle

 goto dongu ‘ Başlangıca (dongu etiketine) geri dön

LEDYak:

 high 1 ‘ P1’e bağlı LED’i yak

 pause 500 ‘ 0,5 sn bekle

 low 1 ‘ P1’e bağlı LED’i söndür

 pause 500 ‘ 0,5 sn bekle

 return ‘ alt-programı sonlandır, pause 1000 komutuna dön

2.12 halt (dur)

a ve b olarak tanımlanmış *iDea kontrol kartı* motor çıkışlarını durdurmak için kullanılır.

Kullanım Şekli

rsBASIC
halt motor

Açıklama

- “motor” a veya b olabilir.
- a motoru *iDea kontrol kartı* P1 ve P2 çıkışlarına bağlanmalıdır.
- b motoru *iDea kontrol kartı* P3 ve P4 çıkışlarına bağlanmalıdır.
- “halt a” komutu low P1, low P2 ile eşdeğerdir.
- “halt b” komutu low P3, low P4 ile eşdeğerdir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
backward a	‘ a motorunun yönünü geri olarak ayarla
motor a,90	‘ a motorunun hızını 90 olarak ayarla
pause 1000	‘ 1 sn bekle
forward a	‘ a motorunun yönünü ileri olarak ayarla
motor a,80	‘ a motorunun hızını 80 olarak ayarla
pause 1000	‘ 1 sn bekle
halt a	‘ a motorunu durdur
pause 1000	‘ 1 sn bekle
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.13 high (yüksek)

Bir sayısal çıkışı yüksek duruma getirmek için kullanılır. Bu duruma getirilen bağlantı +5V çıkış verir.

Kullanım Şekli

rsBASIC
high çıkış

Açıklama

- “çıkış” 1..10 veya 16..17 arasında bir değer olabilir. Bu değerler *iDea kontrol kartı*'nda P1..P10 ve P16..P17 arasındaki bağlantılara karşılık gelmekte olup sayısal giriş/çıkış için ayrılmıştır.
- P1 ve P2 motor a; P3 ve P4 motor b; P5-P10 sayısal çıkışlar; P16 ve P17 iDea kontrol kartı üzerindeki mavi ve kırmızı LED'lerdir.

Örnek

```
dongu:                ‘ Döngü başlangıç etiketi
    high 9             ‘ P9’u aç
    pause 1000        ‘ 1 sn bekle
    low 9              ‘ P9’u kapat
    pause 1000        ‘ 1 sn bekle
    goto dongu        ‘ Başlangıca (dongu etiketine) geri dön
```


2.14 if..then (eğer..ise)

Bir kod parçasının belirli koşullarda çalışmasının istendiği durumlarda kullanılır. if (eğer) tanımından sonraki koşul doğru ise devamındaki bölüm değil ise sonrasındaki kod parçası çalıştırılır.

Kullanım Şekli

rsBASIC
if...then \ elseif...then \ else \ endif
if...then goto (etiket)
if...and/or..then goto (etiket)
if...then gosub
if...and/or...then gosub

Açıklama

- **if .. then** ifadesi arasına sağlanması gereken koşul yazılır. Örneğin, $a > b$, $a = \text{pin1}$, $a = 120$ and $b < 100$.

Örnek

symbol s1 = pin2 ‘ s1 değişkeni *obot P5* girişi olarak tanımlandı
symbol s2 = pin3 ‘ s2 değişkeni *obot P6* girişi olarak tanımlandı

```
if s1=0 and s2=0 then            ‘ eğer s1 ve s2 “0” ise
    high 1                        ‘ 1 numaralı çıkışı aç
elseif s1=1 and s2=0 then       ‘ eğer s1 ve s2 “1” ise
    low 1                         ‘ 1 numaralı çıkışı kapat
else                               ‘ yukarıdaki iki koşulda sağlanmamışsa
    high 1                        ‘ 1 numaralı çıkışı aç
    pause 1000                   ‘ 1 sn bekle
    low 1                         ‘ 1 numaralı çıkışı kapat
endif                             ‘ end..if tanımlaması sona erdi
```

v2.1

2.15 inc (artır)

Sembol ile tanımlanmış bir değeri artırmak için kullanılır.

Kullanım Şekli

rsBASIC
inc sembol

Örnek

symbol i = b0	‘ i değişkeni bayt olarak tanımlandı
i = 2	‘ i değişkenine 2 sabit değeri başlangıçta atandı
dongu:	‘ Döngü başlangıç etiketi
inc i	‘ i değişkenini bir artır
if i = 10 then	‘ if..then..else döngüsü (eğer i = 10 ise)
high 1	‘ 1 numaralı çıkışı aç
else	‘ diğer durumda (eğer i 10’dan farklı ise)
low 1	‘ 1 numaralı çıkışı kapat
endif	‘ end..if tanımlaması sona erdi
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.16 let (ifade tanımla)

İfade tanımlamak ve değişkenler (semboller) üzerinde işlem yapmak için kullanılır.

Kullanım Şekli

rsBASIC
let değişken = sembol
let a = b + 5

Açıklama

– Değişkenler üzerinde aşağıda belirtilen matematiksel işlemler yapılabilir:

- + : toplama
- - : çıkartma
- * : çarpma
- / : bölme
- & : ve
- | : veya
- ^ : xor
- %/ : mod bölümü (kalanı verir)
- |/ : nor
- ^/ : xnor
- inc/dec : artır/azalt

Örnek

symbol i = b0 ‘ i değişkeni bayt olarak tanımlandı
symbol j = b1 ‘ j değişkeni bayt olarak tanımlandı
symbol k = f0 ‘ k değişkeni noktalı sayı olarak tanımlandı

let i = pin1 ‘ i değişkeni pin 1 (P10) değerine eşitlendi
let j = i * 2 ‘ j değişkeni için değer atandı
let k = j / 3 ‘ k değişkenine değer atandı

2.17 low (düşük)

Bir sayısal çıkışı düşük duruma getirmek için kullanılır. Bu duruma getirilen bağlantı 0V çıkış verir.

Kullanım Şekli

rsBASIC
low çıkış

Açıklama

- “çıkış” 1..10 veya 16..17 arasında bir değer olabilir. Bu değerler *iDea kontrol kartı*'nda P1..P10 ve P16..P17 arasındaki bağlantılara karşılık gelmekte olup sayısal giriş/çıkış için ayrılmıştır.
- P1 ve P2 motor a; P3 ve P4 motor b; P5-P10 sayısal çıkışlar; P16 ve P17 iDea kontrol kartı üzerindeki mavi ve kırmızı LED'lerdir.

Örnek

```
dongu:                ' Döngü başlangıç etiketi
                        ' P9'u aç
high 9
pause 1000            ' 1 sn bekle
low 9
pause 1000            ' P9'u kapat
pause 1000            ' 1 sn bekle
goto dongu            ' Başlangıca (dongu etiketine) geri dön
```

2.18 motor

Motorların hızını ayarlamak için kullanılır.

Kullanım Şekli

rsBASIC
motor a,hız motor b,hız

Açıklama

- “hız”, motorun dönme hızını belirleyen değerdir. 0 – 99 değerleri arasında olabilir.
- Motorun dönüş yönünü belirlemek için forward, backward komutları kullanılmalıdır.
- Hız ayarlaması obot kartı üzerinde PWM ile yapılmaktadır. PWM frekansı 30 kHz'tir.
- PWM ile üretilen motor besleme gerilimi iDea kontrol kartı besleme geriliminin % hız değeri olmaktadır. Örneğin 6V pil besleme geriliminde, 80 hız değeri seçildiğinde, motorlara $6V \times \%80 = 4.8V$ uygulanır.

Örnek

dongu:	‘ Döngü başlangıç etiketi
forward a	‘ a motorunu ileri çevir, P1’i aç, P2’yi kapat
motor a, 80	‘ a motorunun hızını 80 olarak ayarla
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.19 pause (bekle)

Belirli bir süre bekleme yapmak için kullanılır. Bu komut çalıştırılırken bekleme zamanının sonuna kadar bir sonraki komut çalıştırılmaz.

Kullanım Şekli

rsBASIC
pause süre

Açıklama

- “süre”, milisaniye olarak bekleme yapılacak değerdir. 0 – 65535 değerleri arasında olabilir.

Örnek

dongu:	‘ Döngü başlangıç etiketi
high 9	‘ P9’u aç
pause 1000	‘ 1 sn bekle
low 9	‘ P9’u kapat
pause 1000	‘ 1 sn bekle
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.20 pulsout (atım gönder)

Sayısal bir çıkıştan belirli uzunluktaki (süredeki) bir atım göndermek için kullanılır. Genellikle RC servo motorların kontrolünde kullanılır.

Kullanım Şekli

rsBASIC
pulsout çıkış, süre

Açıklama

- “çıkış” 5..10 arasında bir değer olabilir. Bu değerler *iDea kontrol kartında* P5..P10 arasındaki bağlantılara karşılık gelmekte olup sayısal giriş/çıkış için ayrılmıştır.
- “süre”, belirtilen süre için çıkış yapılacak bağlantıdaki atımın 10 us’nin (us-mikrosaniye) katı olarak uzunluğunu belirleyen değerdir. Örneğin, 10 olarak verilen süre, 10x10=100 us’dır. 0 – 65535 değerleri arasında olabilir.
- “pulsout” komutu genelde RC Servomotorların denetimi için for..next döngüsü ile birlikte kullanılır.

Örnek

symbol i = b0	‘ i değişkenini bayt olarak tanımladık
dongu:	‘ Döngü başlangıç etiketi
for i = 1 to 100	‘ Belirli sayıda (100) atım üretmek istiyoruz
pulsout 10, 50	‘ P10 çıkışında 0,5ms’lik (500us) bir atım gönder
pause 20	‘ 20 ms bekle
next	‘ i’yi bir artır, gerekli koşul sağlanıyorsa for döngüsüne git
goto dongu	‘ Başlangıca (dongu etiketine) geri dön

2.21 readadc (analog oku)

Analog giriş kanallarından 10 bit çözünürlükte analog sinyal okuma yapmak için kullanılır. Çevrimin gerçekleştirilebilmesi için bu komutun ardından 1 ms bekleme süresi eklenmelidir.

Kullanım Şekli

rsBASIC
readadc giriş, değişken pause 1

Açıklama

- “giriş”, “0” ile “3” arasındaki bir değer olabilir. Bu değerler *iDea kontrol kartında* P11-P14 arasındaki bağlantılara karşılık gelmekte olup analog giriş okumak için ayrılmıştır.
- “değişken” analog sayısal dönüştürme işlemi sonucundaki değer saklandığı kelime sembolüdür. 0-1023 (10bit/0-5V aralığında) değerleri arasında olabilir.

Örnek

```
symbol mesafe = w0          ‘ mesafe değerini saklayacağımız değişken
dongu:                      ‘ Döngü başlangıç etiketi
    readadc 0, mesafe        ‘ P5 analog girişe ait mesafe değerini okuduk
    pause 1                  ‘ 1 ms çevrim için bekle
    if mesafe > 512 then     ‘ mesafe 512’den büyükse (2,5V’tan büyükse)
        forward a           ‘ a motorunu ileri çevir, P1’i aç, P2’yi kapat
    else                     ‘ diğer durumda (mesafe 512’den küçükse)
        backward a          ‘ a motorunu geri çevir, P1’i kapat, P2’yi aç
    endif
goto dongu                  ‘ Başlangıca (dongu etiketine) geri dön
```