به نام خدا

راهنمای ماژول Ev1527_decoder.bas

معرفی

فایل ev1527_decoder.bas یک ماژول است که با استفاده از تایمر 1 در مد کپچر سیگنال چیپ های 1527 یا موارد مشابه را با دقت مناسبی دیکد میکند

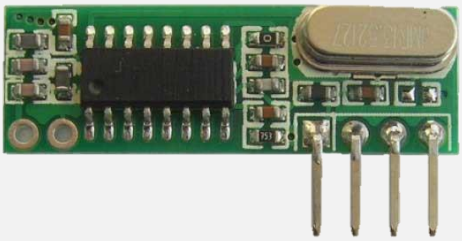
برای جلوگیری از تداخل اسمی حین استفاده از ماژول تمامی متغیر ها، ثابت ها و برچسب های استفاده شده داخل ماژول پیشوند _ev1527d دارند
برای خوانایی بیشتر در این مقاله ، اسم ها بدون پیشوند هستند

در این فایل تنظیماتی در نظر گرفته شده که بر حسب نیاز قابل تغییر هستند. در ادامه راجع به آن ها بیشتر توضیح داده شده.

تنظیمات Prescale تایمر ها به صورت اتوماتیک بر اساس کریستال محاسبه و انتخاب میشود

این ماژول رو به راحتی میتوانید در هر پروژه ای با نوشتن چند خط اضافه کنید

از لحاظ سخت افزاری فقط کافیه خروجی ماژول مدولاتور رو به پایه ICP1 میکرو بدید و تا جای امکان ماژول به میکرو نزدیک باشه ترجیحا Track رو با پلیگان GND شیلد کنید و از مدولاتور های سوپرهترودین استفاده کنید



توضیحات کلی

این ماژول متشکل از چهار مد بوده که بر حسب نیاز قابل استفاده هستند

مد توقف (stop) :
در این مد کلیه تایمر ها و اینتراپت ها غیر فعال و به طور کلی ماژول غیر فعال است.

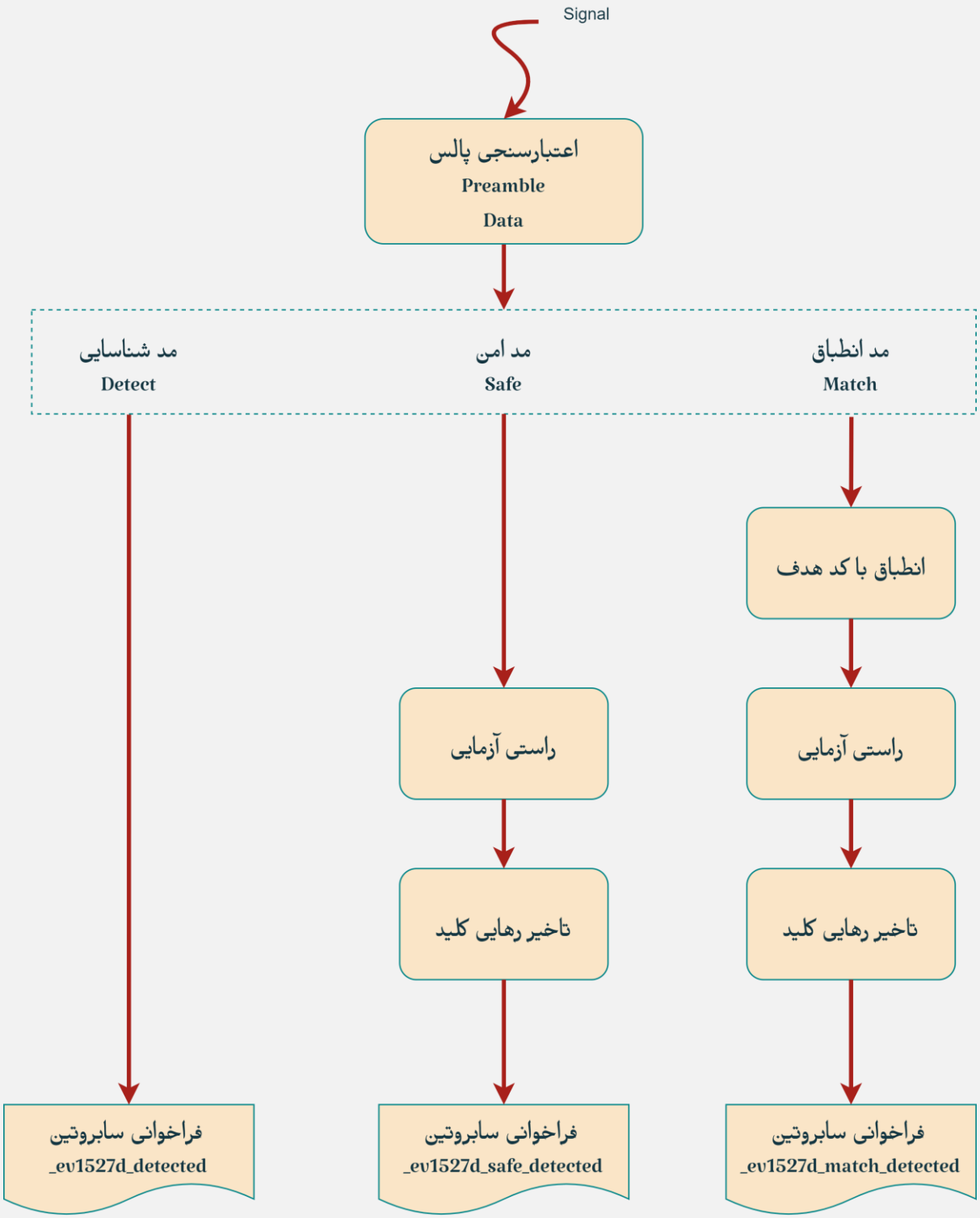
مد شناسایی (detect) :
این مد ساده ترین حالت است و هر بار کدی شناسایی میشود سابروتین detected اجرا و کد دریافتی از طریق آرایه 3 بایتی Code قابل خواندن خواهد بود.

مد امن (safe) :
در این مد راستی آزمایی و تاخیر رها سازی کلید نیز اعمال شده و بعد از گذشتن از این دو فیلتر سابروتین safe_detected فراخوانی و کد دریافتی از طریق متغیر Code قابل خواندن خواهد بود.

مد تطبیق (match) :
این مد زمانی استفاده میشود که ما منتظر کد خاصی هستیم. برای استفاده از این مد ابتدا آرایه 3 بایتی target_code رو مقدار دهی کنید سپس مد رو فعال کنید. در این حالت کد های منطبق بر کد هدف شناسایی شده سپس از فیلتر راستی آزمایی و تاخیر رهاسازی کلید گذشته و نهایتا سابروتین match_detected فراخوانی و کد دریافتی از طریق آرایه 3 بایتی Code قابل خواندن خواهد بود.
لازم به ذکر است که در این مد فقط 20 بیت آدرس target_code جهت تطبیق در نظر گرفته میشوند و مقدار 4 بیت دیتای آخر مهم نیست.

در هر زمانی می توانید از هر مد مستقیما به مد دیگر سوئیچ کنید

_ev1527_decoder.bas



روش استفاده

اضافه کردن ماژول به برنامه:

ماژول رو ابتدای برنامه و بعد از تعریف regfile و crystal اضافه کنید
سه سابروتین detected, safe_detected, match_detected را به انتهای برنامه اضافه کنید

این سابروتین ها حتما باید موجود باشند در غیر اینصورت با خطای کامپایل مواجه میشود

کدی که داخل این سابروتین ها نوشته میشوند باید سریع باشد زیرا تا زمانی که این سابروتین ها به صورت کامل اجرا نشوند عملیات شناسایی دیتا فریم های بعدی آغاز نخواهد شد

```
$regfile = "m32def.dat"
$crystal = 16000000
$baud = 250000

$include "ev1527_decoder.bas"


Do
Loop

_ev1527d_detected:
Return

_ev1527d_safe_detected:
Return

_ev1527d_match_detected:
Return
```

مد شناسایی یا Detect:

برای فعال سازی ماژول در حالت شناسایی کد مطمئن شوید که وقفه ی سراسری فعال باشد سپس start_detect_mode را فراخوانی کنید

در این حالت پس از دریافت هر کد سابروتین detected فراخوانی شده و کد دریافتی از طریق آرایه code قابل خواندن است

متغیر کد یک آرایه 3 بایتی بوده و متشکل از 20 بیت آدرس و 4 بیت دیتا است

در این مد فقط تایمینگ ها بررسی شده و کد از فیلتر راستی آزمایی و تاخیر رهاسازی عبور نمیکند.

```
$regfile = "m32def.dat"
$crystal = 16000000
$baud = 250000

$include "ev1527_decoder.bas"

Relay Alias Portd.2 : Ddrd.2 = 1
Waitms 100
Enable Interrupts
Gosub _ev1527d_start_detect_mode

Do
Loop

_ev1527d_detected:

    If _ev1527d_code(1) = &H12 And _
       _ev1527d_code(2) = &H34 And _
       _ev1527d_code(3) = &H56 _
    Then

       Toggle Relay

    End If

Return

_ev1527d_safe_detected:
Return

_ev1527d_match_detected:
Return
```

با توجه به اینکه کد دریافتی بدون هیچ فیلتری باعث فراخوانی این سابروتین میشود اگر کدی که داخل این سابروتین نوشته میشود به اندازه کافی سریع نباشد (در حد چند میکروثانیه) دیتا فریم بعدی از دست خواهد رفت بنابراین از نوشتن دستوراتی مثل print و wait در این سابروتین خودداری کنید

مد امن یا Safe:

برای فعال سازی ماژول در حالت اسکن کد مطمئن شوید که وقفه ی سراسری فعال باشد سپس سابروتین start_safe_mode را فراخوانی کنید

در این حالت پس از دریافت هر کد سابروتین safe_detected فراخوانی شده و کد دریافتی از طریق آرایه code قابل خواندن است
متغیر کد یک آرایه 3 بایتی بوده و متشکل از 20 بیت آدرس و 4 بیت دیتا است

```
$regfile = "m32def.dat"
$crystal = 16000000
$baud = 250000

$include "ev1527_decoder.bas"

Waitms 100
Enable Interrupts
Gosub _ev1527d_start_safe_mode

Do
Loop

_ev1527d_detected:
Return

_ev1527d_safe_detected:

    Printbin _ev1527d_code(1) ; 3

Return

_ev1527d_match_detected:
Return
```

مد تطبیق یا Match:

برای فعال سازی ماژول در حالت تطبیق
ابتدا متغیر target_code را که یک آرایه 3 بایتی بوده را با کد مورد نظر مقدار دهی کنید

هنگام مقایسه این متغیر فقط 20 بیت آدرس آن در نظر گرفته میشود و مقدار 4 بیت دیتای انتهایی تاثیری ندارد

مطمئن شوید که وقفه ی سراسری فعال باشد سپس سابروتین start_match_mode را فراخوانی کنید

در این حالت آدرس کد دریافتی با آدرس target_code مقایسه می شود و در صورت تطبیق زیر روال match_detected فراخوانی شده

و کد دریافتی از طریق متغیر code که یک آرایه 3 بایتی است قابل خواندن خواهد بود

```
$regfile = "m32def.dat"
$crystal = 16000000
$baud = 250000

$include "ev1527_decoder.bas"

 Waitms 100
 Enable Interrupts

 _ev1527d_target_code(1) = &H12
 _ev1527d_target_code(2) = &H34
 _ev1527d_target_code(3) = &H56

 Gosub _ev1527d_start_match_mode

Do
Loop

_ev1527d_detected:
Return

_ev1527d_safe_detected:
Return

_ev1527d_match_detected:

 Printbin _ev1527d_code(1) ; 3

Return
```

متغیر خواندنی Status

در هر قسمتی از برنامه میتوانید وضعیت ماژول را بررسی کنید
مقدار status می تواند برابر با یکی از ثابت های , stop
detect_mode , safe_mode , match_mode باشد

```
Do

    Select Case _ev1527d_status
       Case _ev1527d_stop:
       Case _ev1527d_detect_mode:
       Case _ev1527d_safe_mode:
       Case _ev1527d_match_mode:
    End Select

Loop
```

تنظیمات

بیشینه و کمینه طول پالس کامل دیتا:

این مقادیر داخل دیتاشیت ev1527 در قالب جدول RoSc&TD آورده شده اند
با توجه به غیر استاندارد بودن بعضی از ریموت ها و برای جلوگیری از عدم شناسایی این ریموت ها این بازه کمی بیشتر در نظر گرفته شده
بقیه فیلتر ها مثل طول پالس preamble ، یک بودن و ... بر اساس این مقادیر حساب خواهند شد

```
'Min/Max of RoSc&TD table on EV1527 datasheet(in micro-secound)
 Const _ev1527d_td_min_us = 1200
 Const _ev1527d_td_max_us = 3200
```

درجه سخت گیری به تایمینگ ها:

مقدار این تابت تعیین میکند که کدام فیلتر ها بر سیگنال دریافتی اعمال شود و می تواند 1، 2 و یا 3 باشد
مقدار 1 فقط طول کامل پالس preamble و طول کامل پالس های دیتا را بررسی میکند
مقدار 2 علاوه بر فیلتر 1 طول یک بودن پالس preamble و پالس های دیتا را نیز بررسی میکند
و مقدار 3 علاوه بر فیتر 2 نسبت کل پالس preamble به یک بودن آن را نیز بررسی میکند که بیشینه کمینه این فیلتر نیز قابل تنظیم است.

مقدار فیلتر هرچه بیشتر باشد چون احتمالا چند دیتا فریم نامعتبر شناخته میشود تا بالاخره یک دیتا فریم ایدآل دریافت شود ممکن است سرعت تشخیص را پایین بیاورد

```
'Timing Strictness (1-3)
'1 -> Checks TPreamble and TData
'2 -> Also checks TPreamble_High and TData_High
'3 -> Also checks TPreamble total to high Ratio
'Default: 1
 Const _ev1527d_timing_strictness = 1
```

بیشینه و کمینه نسبت کل پالس preamble به یک بودن آن:

این تنظیم فقط در صورتی که درجه سخت گیری روی 3 باشد استفاده میشود و بیشینه کمینه نسبت کل پالس preamble به یک بودن آن را مشخص میکند
نسبت ایدآل 32 است اما با توجه به تست هایی که انجام شد به ندرت سیگنال های دریافتی نسبت ایدآل دارند و اکثرا نسبت 25 تا 35 دارند

```
'When (Timing Strictness = 3)
'Min/Max of TPreamble total to high Ratio
 Const _ev1527d_tp2tph_rate_min = 20
 Const _ev1527d_tp2tph_rate_max = 40
```

تعداد راستی آزمایی در مد امن (safe):

تعداد کد هایی که بعد از دریافت کد اول باید دریافت شده و بر کد اول منطبق باشند تا کد دریافتی معتبر شناسایی شود و نهایتا تابع safe_detected را فراخوانی کند
برای مثال اگر مقدار 2 باشد باید حتما 3 فریم با کد یکسان دریافت شود تا سابروتین safe_detected فراخوانی شود
این تنظیم فقط در مد امن استفاده میشود

```
'Number of codes to be received as verification in "Safe Mode"
'Default: 2
 Const _ev1527d_safe_mode_verify_count = 2
```

تعداد راستی آزمایی در مد تطبیق (match):

تعداد کد هایی که بعد از دریافت کد هدف باید دریافت شده و بر کد هدف منطبق باشند تا کد دریافتی معتبر شناسایی شود و نهایتا تابع match_detected را فراخوانی کند
برای مثال اگر مقدار 1 باشد باید حتما 2 فریم با کد یکسان دریافت شود تا تابع match_detected فراخوانی شود

```
'Number of codes to be received as verification in "Match Mode"
'Default: 1
 Const _ev1527d_match_mode_verify_count = 1
```

تاخیر رهاسازی:

در هریک از مد های امن و تطبیق زمانی که کد معتبر دریافت شود برای اینکه زیر روال های مربوطه به صورت مداوم فراخوانی نشوند تاخیر رهاسازی در نظر گرفته شده به این صورت که سابروتین مورد نظر با دریافت کد معتبر یک بار فراخوانی می شوند و برای فراخوانی مجدد
باید کلید رها شده و همچنین زمان تاخیر رهاسازی هم بگذرد
در این صورت با دریافت کد معتبر مجددا سابروتین های مربوطه فراخوانی میشوند
به عبارت دیگر با نگه داشتن یا فشردن و رها کردن پشت سر هم یک کلید سریع تر از این تاخیر تابع فقط یک بار فراخوانی میشود
این تنظیم در مد های امن و تطبیق استفاده میشود و در مد تشخیص کارایی ندارد
در صورتی که این آپشن مقداری بیش از 0 داشته باشد تایمر 2 نیز درگیر و استفاده خواهد شد.
در صورتی که فقط از مد تشخیص استفاده میکنید این مقدار را صفر قرار دهید تا تایمر 2 آزاد شود

```
'After recognizing the code , You need to release the key
'And wait for the release delay to end (in mili-secound)
'When _ev1527d_release_delay_ms is equal to 0, Timer0 is free to use for other purposes
'When _ev1527d_release_delay_ms is more than 0, Timer0 will be used and can't be used for anything else
'Default: 1000
 Const _ev1527d_release_delay_ms = 500
```

این موارد اتوماتیک حساب میشوند و نیازی به تغییر ندارند

```
'** Auto calculation
'** DO NOT CHANGE THEM

'Timer1 calculation
Const _ev1527d_tp_max_s = _ev1527d_td_max_us * 8 / 10 ^ 6
Const _ev1527d_timer1_prescale_min = _ev1527d_tp_max_s * 2 * _xtal / 65536
Const _ev1527d_timer1_prescale = 8 ^ Fix(log(_ev1527d_timer1_prescale_min) / Log(8) + 1)
Const _ev1527d_timer1_frequency = _xtal / _ev1527d_timer1_prescale
Const _ev1527d_timer1_cycles_per_us = _ev1527d_timer1_frequency / 10 ^ 6
Const _ev1527d_timer1_cycles_per_ms = _ev1527d_timer1_frequency / 10 ^ 3

#if _ev1527d_release_delay_ms > 0

   'Timer0 calculation
   Const _ev1527d_timer0_prescale_min = 1 / 10 ^ 3 * _xtal / 256
   Const _ev1527d_timer0_prescale = 8 ^ Fix(log(_ev1527d_timer0_prescale_min) / Log(8) + 1)
   Const _ev1527d_timer0_frequency = _xtal / _ev1527d_timer0_prescale
   Const _ev1527d_timer0_cycles_per_us = _ev1527d_timer0_frequency / 10 ^ 6
   Const _ev1527d_timer0_cycles_per_ms = _ev1527d_timer0_frequency / 10 ^ 3

#endif


'Min/Max of 1 unit of data cycle
Const _ev1527d_tunit_min = _ev1527d_td_min_us * _ev1527d_timer1_cycles_per_us \ 4
Const _ev1527d_tunit_max = _ev1527d_td_max_us * _ev1527d_timer1_cycles_per_us \ 4

'Min/Max of data cycle (in Timer1-Cycle)
Const _ev1527d_td_min = _ev1527d_tunit_min * 4
Const _ev1527d_td_max = _ev1527d_tunit_max * 4
'Min/Max of data "0" cycle timings
Const _ev1527d_t0h_min = _ev1527d_tunit_min
Const _ev1527d_t0h_max = _ev1527d_tunit_max
Const _ev1527d_t0l_min = _ev1527d_tunit_min * 3
Const _ev1527d_t0l_max = _ev1527d_tunit_max * 3
'Min/Max of data "1" cycle timings
Const _ev1527d_t1h_min = _ev1527d_tunit_min * 3
Const _ev1527d_t1h_max = _ev1527d_tunit_max * 3
Const _ev1527d_t1l_min = _ev1527d_tunit_min
Const _ev1527d_t1l_max = _ev1527d_tunit_max
'Min/Max of preamble cycle timings
Const _ev1527d_tp_min = _ev1527d_tunit_min * 32
Const _ev1527d_tp_max = _ev1527d_tunit_max * 32
Const _ev1527d_tph_min = _ev1527d_tunit_min
Const _ev1527d_tph_max = _ev1527d_tunit_max
Const _ev1527d_tpl_min = _ev1527d_tunit_min * 31
Const _ev1527d_tpl_max = _ev1527d_tunit_max * 31
```

متغیر های استفاده شده:

جمعا 36 بایت فضای SRAM استفاده خواهد شد

```
Dim _ev1527d_code(3) As Byte , _ev1527d_target_code(3) As Byte
Dim _ev1527d_buffer(3) As Byte , _ev1527d_previous_code(3) As Byte
Dim _ev1527d_dummy_byte(2) As Byte
Dim _ev1527d_safe_verify_counter As Byte , _ev1527d_match_verify_counter As Byte
Dim _ev1527d_ivalidation As Byte , _ev1527d_ibit As Byte , _ev1527d_ibyte As Byte
Dim _ev1527d_th As Word , _ev1527d_tl As Word , _ev1527d_thl As Word
Dim _ev1527d_t_range(2) As Word , _ev1527d_th_ref As Word
Dim _ev1527d_no_previous_match_ms As Word
Dim _ev1527d_no_target_match_ms As Word
Dim _ev1527d_status As Byte
```

امکان اضافه کردن قابلیت های دیگر هم وجود دارد که بر حسب نیاز بتوان آن ها را فعال یا غیر فعال کرد
مثل:

- یادگیری چندین ریموت و انجام عملیات مقایسه و تطبیق با سرعت بالا
- ذخیره در ایپرام خارجی و داخلی
- سابروتین های رویداد های فشردن کلید و رها کردن کلید به صورت مجزا
- و ...

موفق باشید.