



Subject :

Year .

Month .

Date . ( )

(2) نویسنده در مورد Bounce :

همانطور که میدانیم، کلید در صفحه کیبورد می‌تواند به دلیل مسافت مکانی و غیره، به جای اینکه فقط یک بار کلیک کند، چندین بار کلیک کند. در این حالت، به دلیل مسافت مکانی و غیره، به جای اینکه فقط یک بار کلیک کند، چندین بار کلیک کند.

مسلوبه ها (HWA) می‌توانند به سبب ایجاد حلقه و تداخل نویسی روی بین مربوط شوند. به سبب این، می‌توانند به سبب ایجاد حلقه و تداخل نویسی روی بین مربوط شوند.

عشیره شد چنانچه کلید می‌تواند در صفحه، با فاصله ده، به جای اینکه فقط یک بار کلیک کند، چندین بار کلیک کند. در این حالت، به دلیل مسافت مکانی و غیره، به جای اینکه فقط یک بار کلیک کند، چندین بار کلیک کند.

(1) بستن یک خانوار با کلید

(2) پس از اینکه دکمه را فشار داد، مقدار آن را در نظر داشت Debouncing



Blocking →

هیچ دسترسی قبل از روشن شدن LED یا ISR نیست تنها

Non-Blocking →

محدودیتی ندارد

③ توضیح در keypad

1. keypad: ماینر Constructor: در جاوا شی صفتی که برای ساختن به شماره به سطر و ستون تعداد کلیدها و ماتریس کلیدها

2. getKey: اگر کلیدی فشرده شود این را برمی گرداند. این تابع به صورت Non-Blocking است

3. getKeys: در صورت تغییر وضعیت یکی از کلیدها True در (شار Bool نه char)

4. waitForKey: این تابع به صورت Blocking تا این وضعیت کلیدی همانند در ISR یا می شود

5. getState: وضعیت فعلی کلید از 4 وضعیت (HOLD, IDLE, PRESSED, RELEASED) برمی گرداند

6. keyStateChanged: در صورت تغییر وضعیت کلید True برمی گرداند

Sevan

①- توضیحات ارتباط سریال

این نوع ارتباط سریال بین آردوینو و کامپیوتر یا هر دستگاه جانبی دیگری است. آردوینو با حداقل

یک پورت سریال به نام (UART یا USART) مجهز است. آردوینو Mega مجهز است به

چهار پورت سریال دارد. (0, 1) (2, 3) (4, 5) (6, 7) که برای TX و RX و RX و TX

تفاوتی در ارتباط با کامپیوتر

برای اتصال TX آردوینو را به RX دستگاه جانبی و RX آن را به TX دستگاه متصل می‌کنیم.

⑤ نحوه کار Serial:

1. begin: برای شروع ارتباط استفاده می‌شود و نرخ داده در بیت بر ثانیه (baud) را مشخص می‌کند.

2. end: ارتباط سریال روی پورت را قطع می‌کند و پس از TX و RX در درگاه خروجی کار می‌کند.

3. find: با خواندن دیتا از بافر سریال، در صورتی که داده مورد نظر پیدا شود، True در خروجی پیدا می‌کند.

timeout شد false برمی‌گرداند.

4. parseInt: در صورتی که دیتا به صورت صحیح Valid عددی می‌گردد، در خروجی پیدا می‌کند.

Return می‌کند در غیر این صورت timeout شد صفر برمی‌گرداند.



Subject :

Year .

Month .

Date .

( )

5. `Println` : داده خروجی را به شکل قرارداد ASCII به صورت قابل خواندن بودن توسط این

نمونه می کند. سه ایزال یک کاراکتر با پسوند ('r' یا ASCII B) و یک کاراکتر خط جدید

('n' یا ASCII 10) چاپ می کند.

6. `Read` : داده ورودی سریال را می خواند.

7. `readStringUntil` : کاراکترهای ورودی سریال را تا کاراکتر terminator می خواند و به صورت رشته

برمی گرداند. در صورتی که terminator نیامد تا timeout شود به خواندن ادامه می دهد. تا باغ

`SetTimeout` هم می تواند برای آن محدودیت تعیین کرد.

8. `Write` : داده ها را به صورت تایم یا چندتا در پورت سریال می نویسد (مشاره به نوشتن

کاراکترها می باشد) نمونه یک (عبر به صورت `String`) از `Print` استفاده می شود

Subject :

مقایسه UART , USART :

(Universal Asynchronous Receiver/Transmitter) UART

یکی از انواع جانبی میکروکنترلرهاست که جابجایی داده‌ها و ورودی و خروجی را به صورت سریال می‌تواند انجام دهد.

این دستگاه یک بیت شروع و یک بیت پایانی را در هر انتقال داده قرار می‌دهد.

(Universal Sync/Async Receiver/Transmitter) USART

یکی از انواع جانبی میکروکنترلرهاست که به دو روش (Sync) و (Async) می‌تواند داده‌ها را با UART مبادله کند.

سرعت آن بیشتر از UART است، تفاوت هم در یک بیت استقراری است.