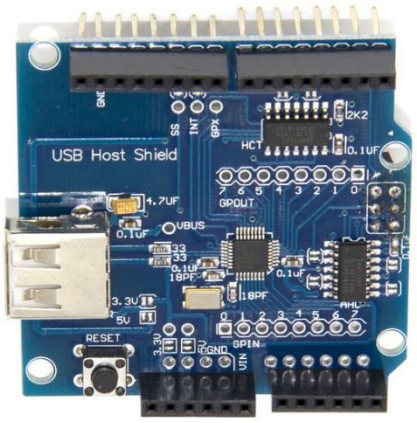
1420991501
VÝROBNÍ ČÍSLO

USB host shield

1. POPIS



Jedná se o shield přímo kompatibilní s deskami Arduino Uno, Leonardo, NHduino nebo Mega 2560. Shield umožňuje z Arduina pomocí čipu MAX3421E vytvořit hostitelské USB zařízení. USB host shield podporuje hardware jako např. myš, klávesnice, gamepad (PS3, Wii, Xbox 360) a další. Modul je schopen pracovat také s chytrými telefony, digitálními fotoaparáty, GPS moduly, paměťovými zařízeními (čtečky paměťových karet nebo USB flash disky) a USB Bluetooth moduly.

Základní charakteristika:

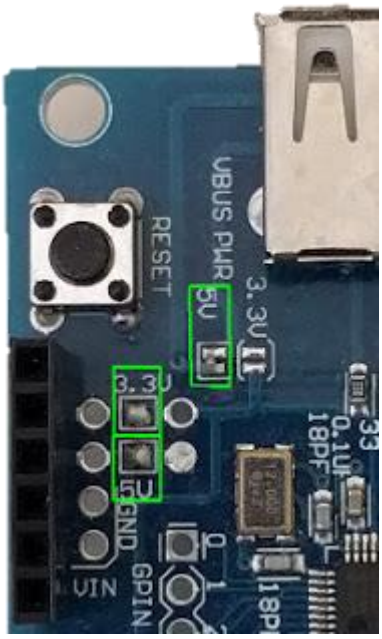
- USB vstup
- čip MAX3421E
- podpora periferií, chytrých telefonů, fotoaparátů a dalších USB zařízení
- kompatibilita s platformami Arduino Uno (a jemu podobné) nebo Mega 2560

2. SPECIFIKACE

Hlavní čip	MAX3421E	**Komunikační rozhraní**	SPI
Vstupní konektor	USB typ A (samice)	**Rozměry (mm)**	55 x 53
Přímá kompatibilita	Arduino Uno a Mega 2560	**Hmotnost**	15 g

3. ZAPOJENÍ

Shield je určen k zasunutí přímo do kompatibilních vývojových desek. Nemá však propojené pájecí můstky vyznačené na obrázku zelenou barvou. Tyto můstky je nutné pro správnou funkci shieldu propojit.



4. UKÁZKA PROGRAMU

K správnému zkompilování kódu jsou za potřeby knihovny USB_Host_Shield_2.0-1.3.2 a spi4teensy3. Kód je převzat z příkladů knihovny určené pro USB host shield. Kód čte pohyby myši po kartézské soustavě souřadnic a zachycuje stisky tlačítek.

```
#include <hidboot.h>
#include <usbhub.h>
// Satisfy the IDE, which needs to see the include statment in the ino too.
#ifdef dobogusinclude
#include <spi4teensy3.h>
#endif
#include <SPI.h>

class MouseRptParser : public MouseReportParser
{
protected:
        void OnMouseMove        (MOUSEINFO *mi);
        void OnLeftButtonUp     (MOUSEINFO *mi);
        void OnLeftButtonDown (MOUSEINFO *mi);
        void OnRightButtonUp    (MOUSEINFO *mi);
        void OnRightButtonDown          (MOUSEINFO *mi);
        void OnMiddleButtonUp (MOUSEINFO *mi);
        void OnMiddleButtonDown         (MOUSEINFO *mi);
};
void MouseRptParser::OnMouseMove(MOUSEINFO *mi)
```

```
{
  Serial.print("dx=");
  Serial.print(mi->dX, DEC);
  Serial.print(" dy=");
  Serial.println(mi->dY, DEC);
};
void MouseRptParser::OnLeftButtonUp            (MOUSEINFO *mi)
{
  Serial.println("L Butt Up");
};
void MouseRptParser::OnLeftButtonDown          (MOUSEINFO *mi)
{
  Serial.println("L Butt Dn");
};
void MouseRptParser::OnRightButtonUp           (MOUSEINFO *mi)
{
  Serial.println("R Butt Up");
};
void MouseRptParser::OnRightButtonDown         (MOUSEINFO *mi)
{
  Serial.println("R Butt Dn");
};
void MouseRptParser::OnMiddleButtonUp          (MOUSEINFO *mi)
{
  Serial.println("M Butt Up");
};
void MouseRptParser::OnMiddleButtonDown        (MOUSEINFO *mi)
{
  Serial.println("M Butt Dn");
};

USB     Usb;
USBHub     Hub(&Usb);
HIDBoot<USB_HID_PROTOCOL_MOUSE>    HidMouse(&Usb);
MouseRptParser  Prs;

void setup()
{
  Serial.begin( 115200 );
#if !defined(__MIPSEL__)
  while (!Serial); // Wait for serial port to connect - used on Leonardo, Teensy and other boards with built-in USB CDC serial connection
#endif
  Serial.println("Start");

  if (Usb.Init() == -1)
    Serial.println("OSC did not start.");

  delay( 200 );
  HidMouse.SetReportParser(0, &Prs);
}

void loop()
{
 Usb.Task();
}
```