PS4 – STM32 remote control Lib simple tutorial

William Leung

Before start, you need:

1. Arduino pro mini
2. mini usb host shield 2.0
   ➔ or another similar planform e.g. Arduino UNO with Arduino usb host shield
3. NUCLEO-F103RB
   ➔ Or other development board support Mbed OS
   ➔ you can try implement the library to another platform, not difficult
4. Bluetooth adaptor
5. PS4 (dualshock 4) wireless controller
6. All program code (form Robotic_share/Electronics/william/PS4remote/tutorial set)

Step 1.

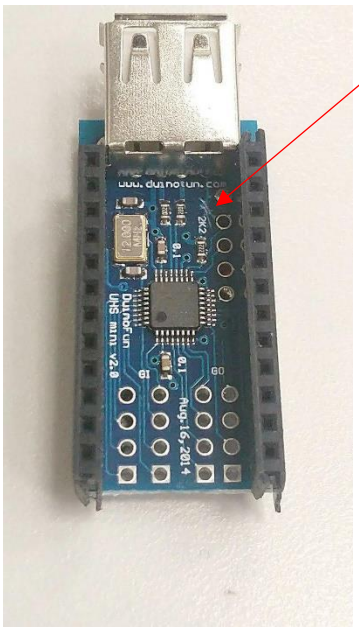
Change the routing of the mini usb host shield 2.0.

according to https://www.circuitsathome.com/usb-host-shield-hardware-manual/

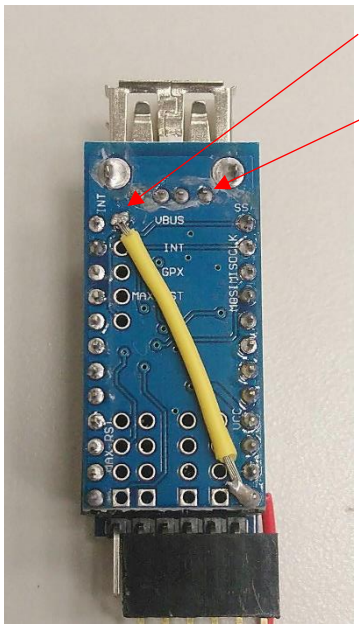
if we want to have a 5V usb power instead of 3.3V, we need to modify the routing.

Detail step:

Power the board with 5V on RAW pin, cut the trace inside VBUS jumper, provide 5V from RAW to VBUS and get 3.3V from Arduino to the shield in the usual manner via 3.3V pin. While making this mod be extra careful and don't short VBUS pad to USB connector shield.



cut it.

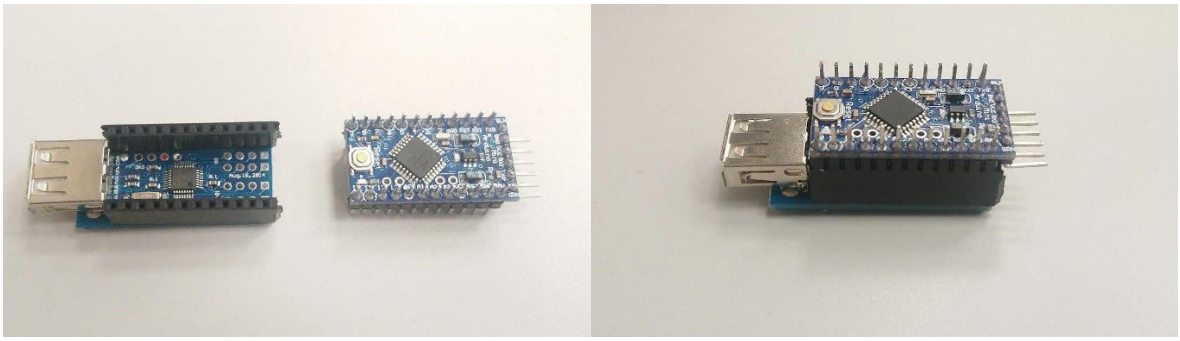


Connect the RAW to here.

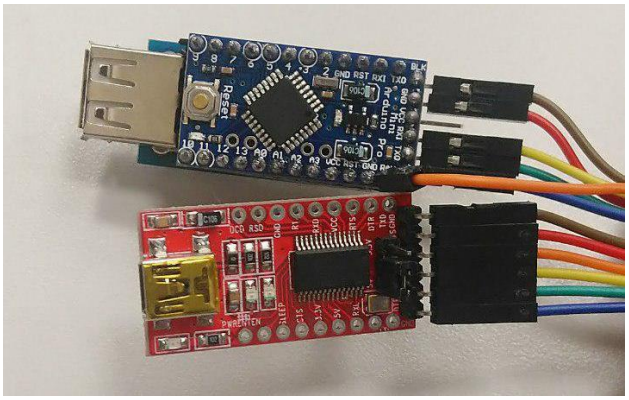
Or here also ok.

Step 2.

Connect the Arduino mini with the mini usb host shield 2.0.



Burn program using the FTDI FT232RL or other USB to TTL module. Remember connecting the 5V power to the RAW pin.



Try to burn the usb mini host shield 2.0 example PS4BT.ino. you can get it form ardunio library manager, more detail in https://github.com/felis/USB_Host_Shield_2.0#arduino-library-manager.

At the first time you need to hold down the Share button and then hold down the PS without releasing the Share button. The PS4 controller will then start to blink rapidly indicating that it is in pairing mode.

If you success to pair up the PS4 controller, the LED light should stay in blue, and the serial monitor will showing the button information.

After success the pair up you can burn my Arduino code into the pro mini.

The file should be PS4_lib_BT /PS4_lib_BT.ino

Step 3.

Connect the Arduino pro mini UART to STM32 UART1.

(My setting: Arduino TX0 <-> STM PA_10, Arduino RX1 <-> STM PA_9).

Using Mbed online compiler open my library: STM_PS4_Lib_10_3.zip.

You can write you code in PS4_Commnad.cpp

The library has 4 set of function like the picture show in left.

high_func_XXX -> mean when the button press, it will loop the code inside.

low_func_XXX -> mean when the button keeps release, it will loop the code inside.

Fall_func_XXX -> when the button state from press change to release it will call once.

rise_func_XXX -> when the button state from release change to press it will call once.

```
void    high_func_CommandChange(){}
void    high_func_LH(){}
void    high_func_RH(){}
void    high_func_L2(){}
void    high_func_R2(){}
void    high_func_TRI(){}
void    high_func_CIR(){}
void    high_func_CRO(){}
void    high_func_SQU(){}
void    high_func_UP(){}
void    high_func_RIGHT(){}
void    high_func_DOWN(){}
void    high_func_LEFT(){}
void    high_func_L1(){}
void    high_func_L3(){}
void    high_func_R1(){}
void    high_func_R3(){}
void    high_func_SHARE(){}
void    high_func_OPTIONS(){}
```

The library already does an example using the UP button, you can try the up button and monitor the STM serial output to understand the looping behavior.

More Detail of the Library

Analog button

In the above picture, you can see that LH,RH,L2,R2 have some value after button. Such button is an analog button, for example the LH have a X,Y value to represent the level of the switch.

At the library, the value also saved for programming.
you can use the function below to get the value.

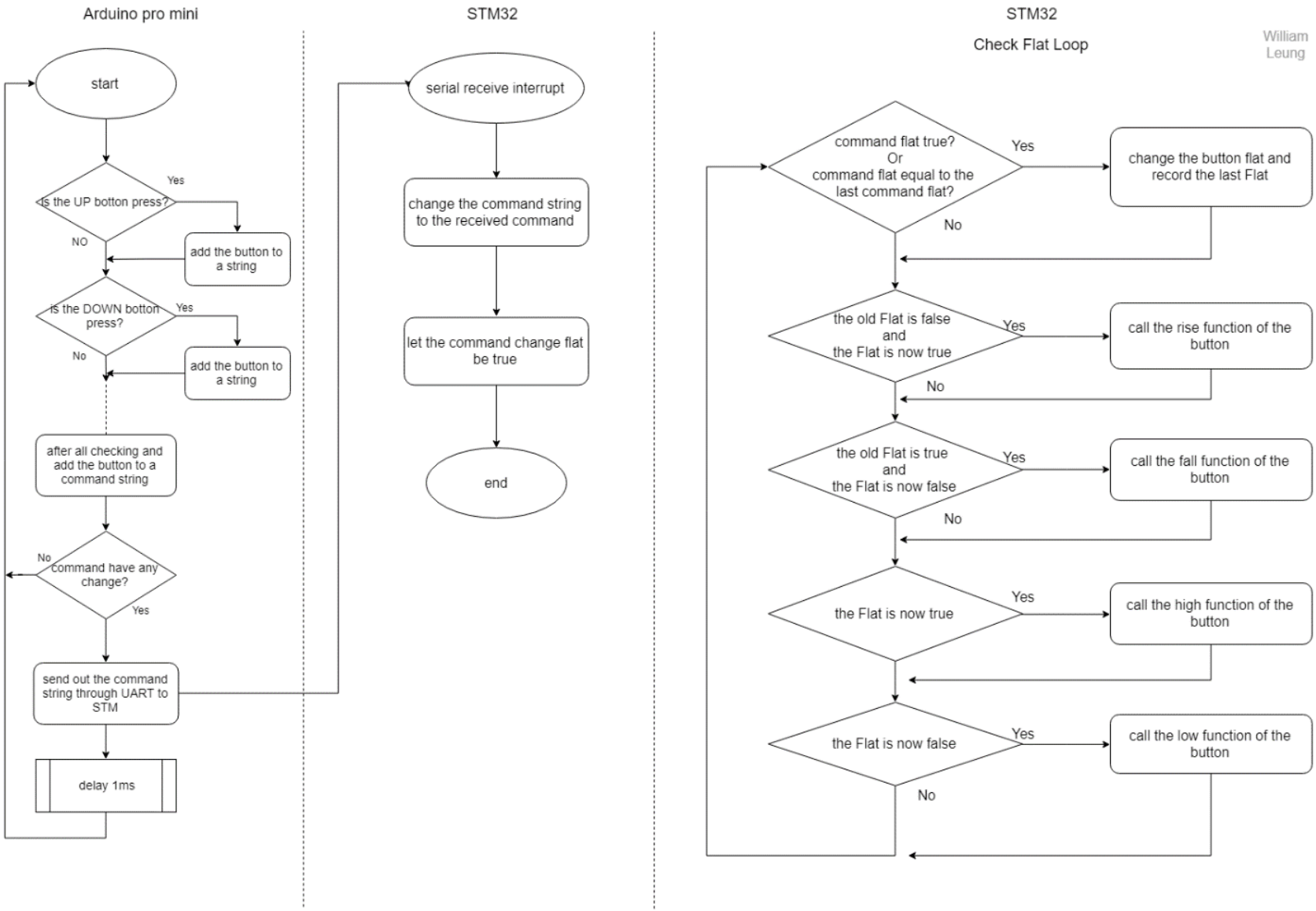
```
int getLX();
int getLY();
int getRX();
int getRY();
int getLA2();
int getRA2();
```

Double button

Because there are too many button combinations. If you want to do some function when two buttons press as same time. You need to add some condition at the checkflatloop(), and need to understand the looping logic. You can follow the flow chat and the code to further develop the Library.

looping efficiency

In the checkflatloop(), it will always call some function. For example, when you connect the remote, but no button is pressed, it will also be looping all low button function. If you want to get a higher efficiency. You can modify the checkflatloop() to disable the low, high condition or just don't call the function pointer.


Arduino pro mini
STM32
start
Is the UP botton press?
Yes
NO
add the button to a string
is the DOWN botton press?
Yes
No
add the button to a string
after all checking and add the button to a command string
command have any change?
No
Yes
send out the command string through UART to STM
delay 1ms
serial receive interrupt
change the command string to the received command
let the command change flat be true
end
STM32
Check Flat Loop
William Leung
command flat true? Or command flat equal to the last command flat?
Yes
No
change the button flat and record the last Flat
the old Flat is false and the Flat is now true
Yes
No
call the rise function of the button
the old Flat is true and the Flat is now false
Yes
No
call the fall function of the button
the Flat is now true
Yes
call the high function of the button
the Flat is now false
Yes
No
call the low function of the button