Results 1 to 4 of 4

Thread: Gamepad emulation using teensy 4.0

11-01-2020, 12:52 PM #1

marcob

Member

Join Date: Sep 2020

Posts: 77

Gamepad emulation using teensy 4.0

Hi guys.

Is it possible to emulate a gamepad using a teensy 4.0 or only a specific teensy model can emulate a gamepad?

I am looking into making xinput controller using the teensy 4.0, same as it has been done here with a different microcontroller

https://www.partsnotincluded.com/how...rduino-xinput/

Thanks

11-01-2020, 03:16 PM #2

MichaelMeissner

Senior Member+

Join Date: Nov 2012

Location: Ayer Massachussetts

Posts: 3,897

> Originally Posted by **marcob**
>
> Hi guys.
>
> Is it possible to emulate a gamepad using a teensy 4.0 or only a specific teensy model can emulate a gamepad?
>
> I am looking into making xinput controller using the teensy 4.0, same as it has been done here with a different microcontroller
>
> https://www.partsnotincluded.com/how...rduino-xinput/
>
> Thanks

It depends on the code, and what devices the code uses. There are a few things that might prevent it from working on a Teensy 4.0/4.1. Many of things are minor, and can be changed in the code:

- The define for the Teensy 4.0/4.1 processor is different from the older Teensies. So if the code only explicitly checks for a Teensy LC, 3.2, 3.5, or 3.6, it might miss the Teensy 4.0 or 4.1. If it checks the higher level define (**CORE_TEENSY**), it may be fine.
- Only the first serial port is in the same location in all of the Teensy LC, 3.x, or 4.x. If the code uses Serial2 or Serial3, then you would need to use different pins. Given it also runs on AVR processors like the Leonardo that only have one serial port, I suspect it may not be an issue.
- If it uses a DAC (digital -> analog converter) to produce sounds, that will be a problem on the Teensy 4.0/4.1, which does not have a DAC. There are various ways to do sound on the Teensy 4.0/4.1, but it will involve re-coding the parts, and perhaps some trade-offs in terms of pin selection.
- If it uses analog pins A10 and A11 (on the inside of the Teensy LC, 3.x), these pins aren't in that position in the Teensy 4.0/4.1 (and on the 4.0, A10/A11 are on solder pads underneath the Teensy).
- If it uses the analog reference pin (AREF), the Teensy 4.0/4.1 doesn't have an AREF. If you are using AREF, but not setting it to an explicit value, you can generally use 3.3v as AREF.
- The code appears to have its own **boards.txt** file to add options. You would need to add equivalent options for the Teensy 4.0/4.1.

Reply With Quote

11-01-2020, 09:01 PM #3

marcob

Member
Join Date: Sep 2020
Posts: 77

> Originally Posted by **MichaelMeissner**
>
> It depends on the code, and what devices the code uses. There are a few things that might prevent it from working on a Teensy 4.0/4.1. Many of things are minor, and can be changed in the code:
>
> - The define for the Teensy 4.0/4.1 processor is different from the older Teensies. So if the code only explicitly checks for a Teensy LC, 3.2, 3.5, or 3.6, it might miss the Teensy 4.0 or 4.1. If it checks the higher level define (**CORE_TEENSY**), it may be fine.
> - Only the first serial port is in the same location in all of the Teensy LC, 3.x, or 4.x. If the code uses Serial2 or Serial3, then you would need to use different pins. Given it also runs on AVR processors like the Leonardo that only have one serial port, I suspect it may not be an issue.

- If it uses a DAC (digital -> analog converter) to produce sounds, that will be a problem on the Teensy 4.0/4.1, which does not have a DAC. There are various ways to do sound on the Teensy 4.0/4.1, but it will involve re-coding the parts, and perhaps some trade-offs in terms of pin selection.
- If it uses analog pins A10 and A11 (on the inside of the Teensy LC, 3.x), these pins aren't in that position in the Teensy 4.0/4.1 (and on the 4.0, A10/A11 are on solder pads underneath the Teensy).
- If it uses the analog reference pin (AREF), the Teensy 4.0/4.1 doesn't have an AREF. If you are using AREF, but not setting it to an explicit value, you can generally use 3.3v as AREF.
- The code appears to have its own **boards.txt** file to add options. You would need to add equivalent options for the Teensy 4.0/4.1.

Thank you for the reply.
I see. So it would be not that easy for novice like me

The reason why I am asking about the 4.0 is because it has two i2c channels, and I need that for the project I have been doing and shared here on the forum

Reply With Quote

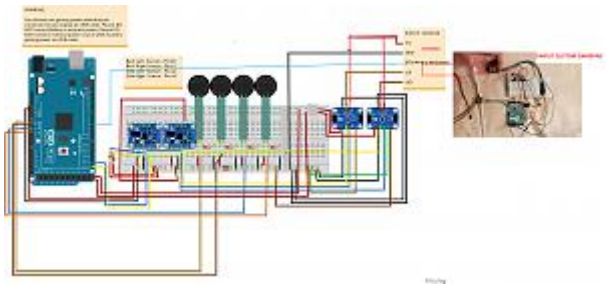
11-01-2020, 10:17 PM #4

marcob

Member
Join Date: Sep 2020
Posts: 77

I would like to share a method I was using when my system was running on the mega.
Maybe I can do the same with the teensy 4.0

I can actually connect the teensy 4.0 to the leonardo by replacing the joystick hooked on the leonardo.
I wish the code of the xinput controller would be updated for the teensy 4.0 of course.



code for the leonardo

Code:

```
// Set left joystick
if (UseLeftJoystick == true) {
  int leftJoyX = analogRead(Pin_LeftJoyX);
  int leftJoyY = analogRead(Pin_LeftJoyY);

  // White lie here... most generic joysticks
  // inverted by default. If the "Invert" va
  // then we need to do this transformation.
```

```
    if (InvertLeftYAxis == false) {
      leftJoyY = ADC_Max - leftJoyY;
    }

    XInput.setJoystick(JOY_LEFT, leftJoyX, lef
  }

  // Set right joystick
  if (UseRightJoystick == true) {
    int rightJoyX = analogRead(Pin_RightJoyX);
    int rightJoyY = analogRead(Pin_RightJoyY);

    if (InvertRightYAxis == false) {
      rightJoyY = ADC_Max - rightJoyY;
    }

    XInput.setJoystick(JOY_RIGHT, rightJoyX, r
  }

  // Send control data to the computer
  XInput.send();
}
```

My old Code for the mega

Code:

```
        angle = 78;
        press2 = 1;
      }
      else {
        //servo1.write(78 , 20);
        ds3502_2.setWiperDefault(50);
        angle = 78;
        press2 = 1;
      }
      freq++;
      runC = 1;
    }
    else if ((step == 1 || step == 2 &&  millis
      //servo1.write(78 , 20);
      ds3502_2.setWiperDefault(50);
      angle = 78;
      freq = 1;
      press2 = 1;
    }
    timer = millis();
  }

  if (press1 == 1 || press2 == 1)
    digitalWrite(Button , LOW);
  else
    digitalWrite(Button , HIGH);

  Serial.println("Button: " + String(digitalRea
  laststep = steps;
  delay(80);
}
```

and I also made the xinput controller wireless using a minirouter and virtualhere, same as i did for this psaim controller that I made wireless connected to the pc

https://streamable.com/t4599

Reply With Quote

Quick Navigation **Technical Support & Questions** **Top**

« Previous Thread | Next Thread »

Posting Permissions

You may not post new BB code is On

You may not post new threads
You may not post replies
You may not post attachments
You may not edit your posts

BB code is On
Smilies are On
[IMG] code is On
[VIDEO] code is On
HTML code is Off

Forum Rules