XInput Gamepad

Introducere

Proiectul reprezinta un gamepad compatibil cu Windows prin XInput.

XInput este un API prin care programele pot comunica cu controllere Xbox. Asadar, proiectul vizeaza implementarea unui gamepad/controller care sa emuleze un controller de Xbox.

Descriere generală

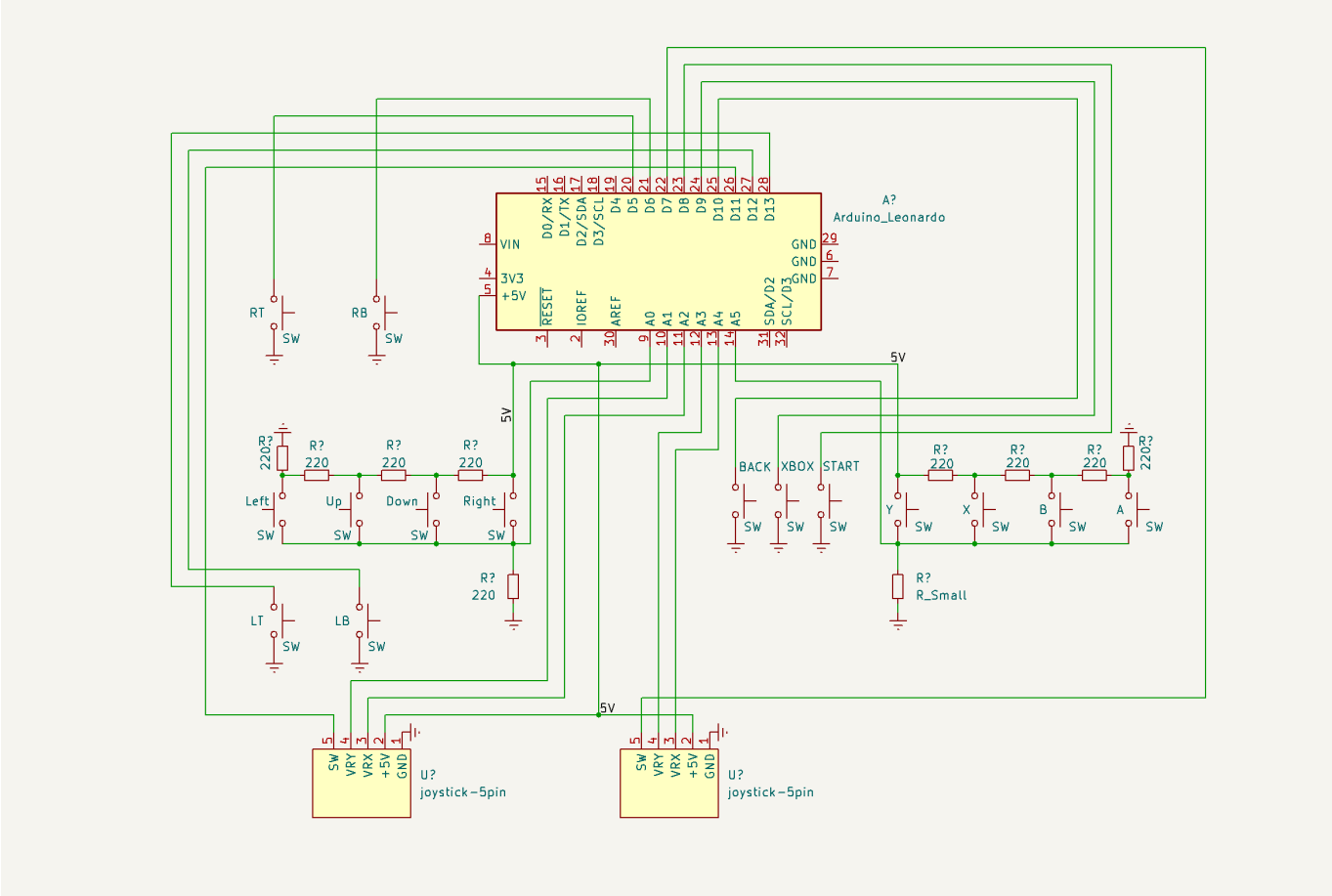
Pentru a putea respecta standardul impus de XInput, este nevoie de o platforma cu suport nativ pentru USB si de perifericele corespunzatoare unui controller de Xbox: butoanele A/B/X/Y, D-pad, doua joystick-uri, butoanele RB/LB/RT/LT si butoanele de meniu (Back, Start, Xbox).

Hardware Design

Lista de piese

- Arduino Leonardo
- 15 butoane
- 2 joystick-uri
- 10 rezistori de 220ohm
- Jumpere
- ~~Stripboard~~ Breadboard
- Cablu micro USB

Schema Electrica



Pentru implementarea proiectului este nevoie de o placuta cu port USB nativ, cum este Arduino Leonardo.

Am ales sa folosesc doua divizoare de tensiune pentru a economisi pinii digitali de pe placuta. Butoanele directionale si butoanele A/B/X/Y sunt grupate cate 4 in cate un divizor de tensiune.

Dezavantajul acestor grupari, este ca nu se pot citi doua sau mai multe butoane, din acelasi grup, simultan. De exemplu, apasarea simultana a butoanelor A si B, v-a trimite pe pinul analogic valoarea corespunzatoare apasarii butonului B, nu a amandurora.

Un alt compromis il constituie faptul ca, din motive de disponibilitate, butoanele “Trigger” nu sunt analogice. Valoarea trimisa la API-ul XInput este fie 0% apasat, fie 100% apasat.

Software Design

Pentru dezvoltare am folosit ArduinoIDE, si biblioteca ArduinoXInput.

- XInput USB Core for Arduino AVR
- Arduino XInput Library

In momentul in care se incarca un sketch pe o placuta cu XInput (Tools→Board→XInput Boards) este OBLIGATORIU ca atunci cand apare “Uploading...” in ArduinoIDE, sa apasati de doua ori pe butonul de reset de pe placuta.

Folosirea acestor biblioteci reduce semnificativ dificultatea implementarii programului. Mai intai setez constantele pinilor

```
const uint8_t abxy_pin = A5;
const uint8_t yL_pin = A1;
const uint8_t xL_pin = A2;
const uint8_t yR_pin = A3;
const uint8_t xR_pin = A4;
const uint8_t dpad_pin = A0;


const uint8_t lt_pin = 13;
const uint8_t lb_pin = 12;
const uint8_t swL_pin = 11;
const uint8_t view_pin = 10;
const uint8_t xbox_pin = 9;
const uint8_t home_pin = 8;
const uint8_t swR_pin = 7;
const uint8_t rb_pin = 6;
const uint8_t rt_pin = 5;
```

Valorile joystick-urilor sunt de 16 biti cu semn, dar ADC-ul de pe placuta are o rezolutie de doar 10 biti. Din fericire, biblioteca XInput ofera posibilitatea de a seta un range maxim pentru aceste valori.

```
const uint16_t analog_range = 1023;

void setup() {
  // ...
  XInput.setRange(JOY_LEFT, 0, analog_range);
  XInput.setRange(JOY_RIGHT, 0, analog_range);
}
```

Initializez biblioteca

```
void setup() {
  // ...
  XInput.begin();
}
```

Functiile bibliotecii folosesc urmatoarele constante pentru butoanele emulate:

```
enum XInputControl : uint8_t {
        BUTTON_LOGO = 0,
        BUTTON_A = 1,
        BUTTON_B = 2,
        BUTTON_X = 3,
        BUTTON_Y = 4,
        BUTTON_LB = 5,
        BUTTON_RB = 6,
        BUTTON_BACK = 7,
```

```
        BUTTON_START = 8,
        BUTTON_L3 = 9,
        BUTTON_R3 = 10,
        DPAD_UP = 11,
        DPAD_DOWN = 12,
        DPAD_LEFT = 13,
        DPAD_RIGHT = 14,
        TRIGGER_LEFT = 15,
        TRIGGER_RIGHT = 16,
        JOY_LEFT,
        JOY_RIGHT,
};
```

Pentru interfatarea cu API-ul XInput folosesc urmatoarele functii puse la dispozitie de biblioteca:

```
XInput.setButton(XInputControl button, bool state);

// Seteaza starea butoanelor dpad-ului la valorile parametrilor
XInput.setDpad(bool up, bool down, bool left, bool right);

// Seteaza valoarea Joystick-ului la valoarea celui de-al doilea parametru.
// Daca invert este true, atunci valorile de pe axa respectiva sunt
inversate
XInput.setJoystickX(XInputControl joy, int32_t x, bool invert);
XInput.setJoystickX(XInputControl joy, int32_t y, bool invert);

void setTrigger(XInputControl trigger, int32_t val);
```

Pentru toate butoanele legate la pini digitali apelez XInput.setButton().

Valorile analogice (Dpad si ABXY) sunt parsate in functie de nivelurile de tensiune date de divizor.

```
void parseVoltageDivider(const int value, bool& int1, bool& int2, bool& int3,
bool& int4)
{
  if (value > 15 && value < 210)        int1 = true;
  else if (value >= 210 && value < 290) int2 = true;
  else if (value >= 290 && value < 850) int3 = true;
  else if (value >= 850)                int4 = true;
}
```

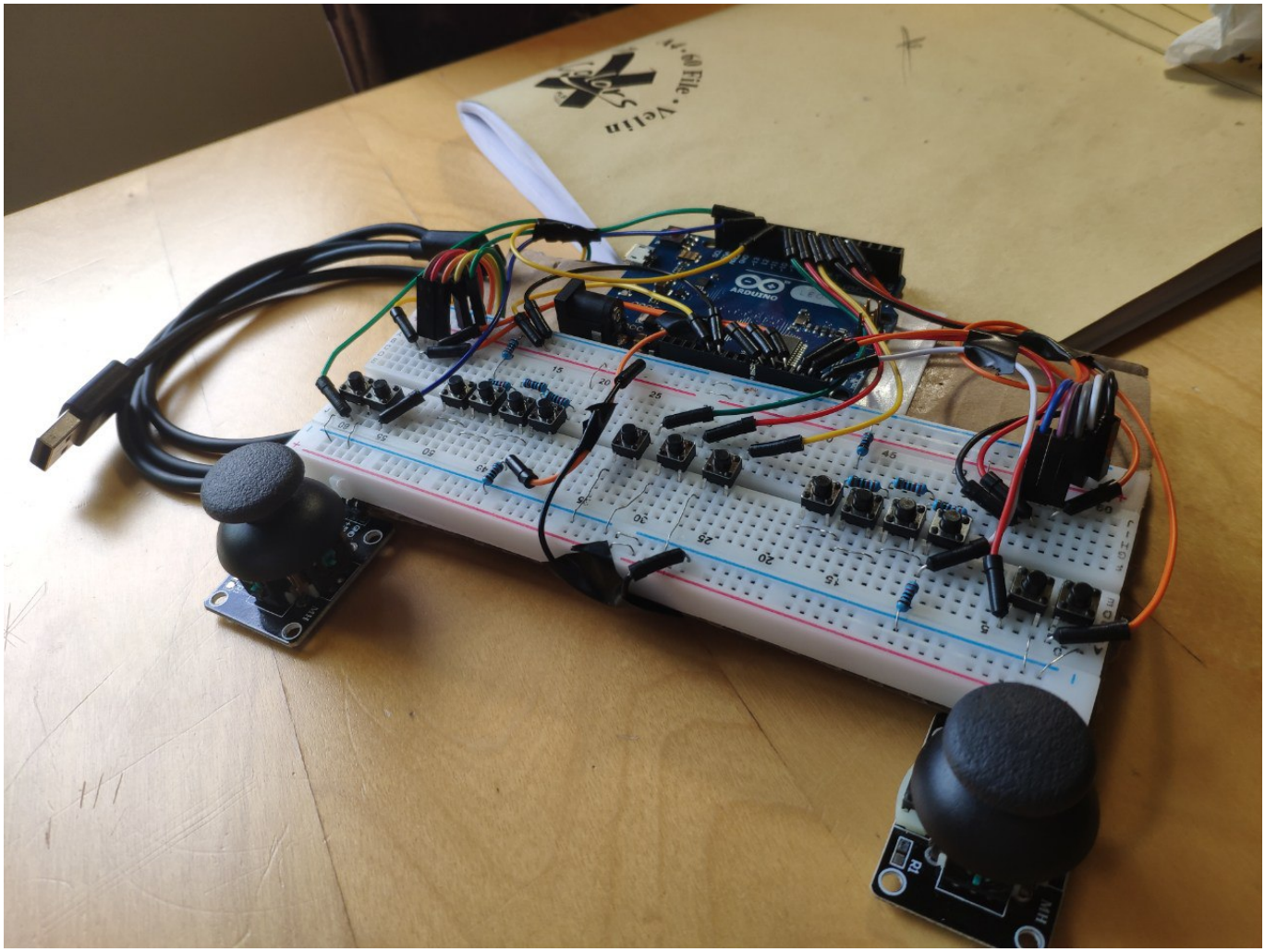
Butoanele trigger sunt setate folosind urmatoarea functie:

```
void setTrig(bool button, uint8_t XIControl)
{
  if (button) {
    /* Set 100% */
    XInput.setTrigger(XIControl, 255);
  } else {
    /* Release */
    XInput.setTrigger(XIControl, 0);
```

```
    }
}
```

Rezultate Obţinute

Am reusit sa implementez tot ce mi-am propus. Mi-ar fi placut sa pot lipi proiectul pe un perfboard, dar nu prea m-am descurcat.



Concluzii

Proiectul functioneaza in totalitate, este detectat de Windows si tratat ca un controller de Xbox. Se pot aduce imbunatatiri la rigiditatea constructiei si la felul in care sunt legate butoanele, astfel incat sa se poata citi simultan butoane din grupurile Dpad si ABXY.

Jurnal

29.05.2022 – Finalizare documentatie 02.06.2022 – Export to pdf

Download

gamepadxinput_sketch.zip

Bibliografie/Resurse

Resurse Software

- XInput USB Core for Arduino AVR
- Arduino XInput Library

Resurse Hardware

- Divizor de tensiune

Export to PDF