Visualization of Human Posture Based on Accelerometer Data

Evgeny Tsvetkov, Alexander Borodin

Petrozavodsk State University
Department of Computer Science

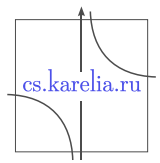
This project is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland

11th FRUCT conference

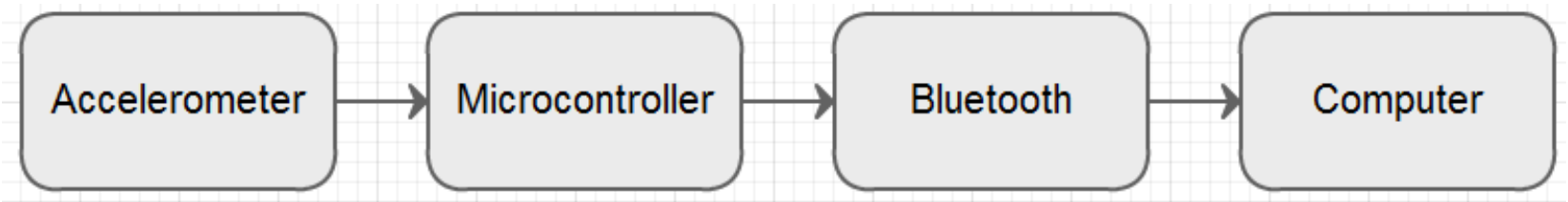
April 23–27, Saint-Petersburg, Russia

cs.karelia.ru

Table of Contents

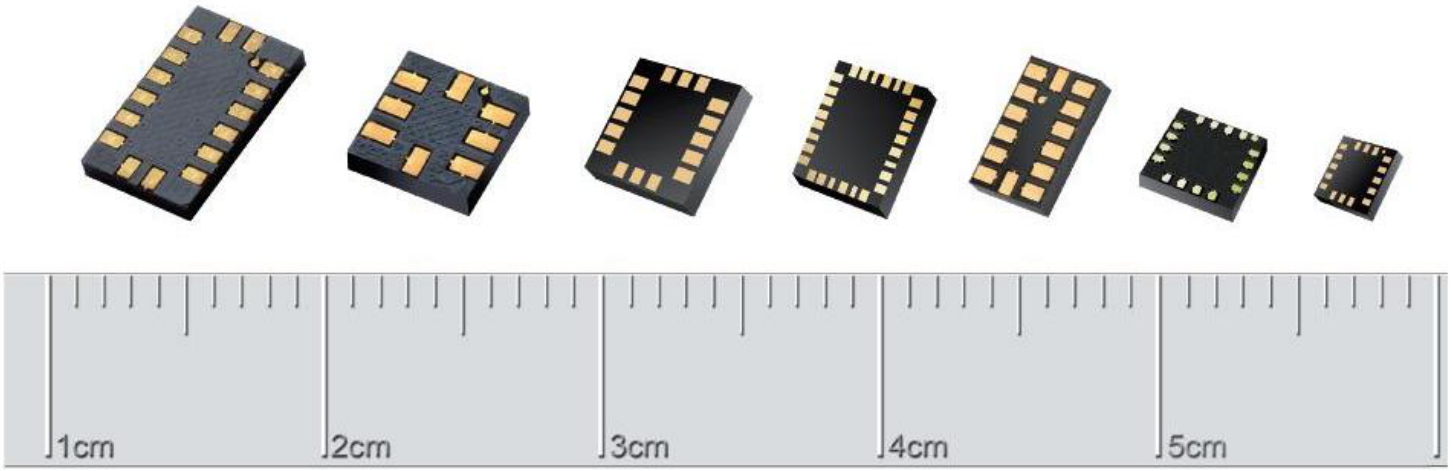


Architecture



cs.karelia.ru

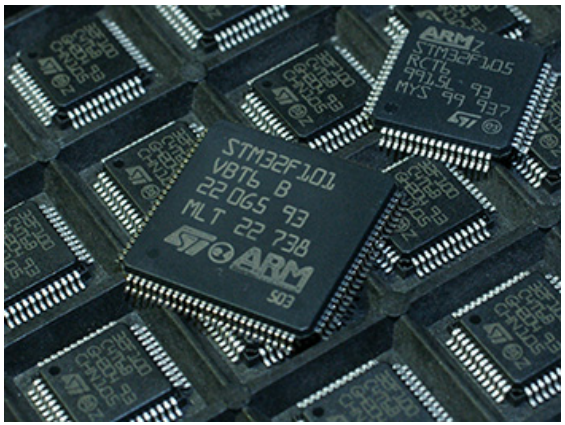
Choice of sensor



- analog output
- digital output

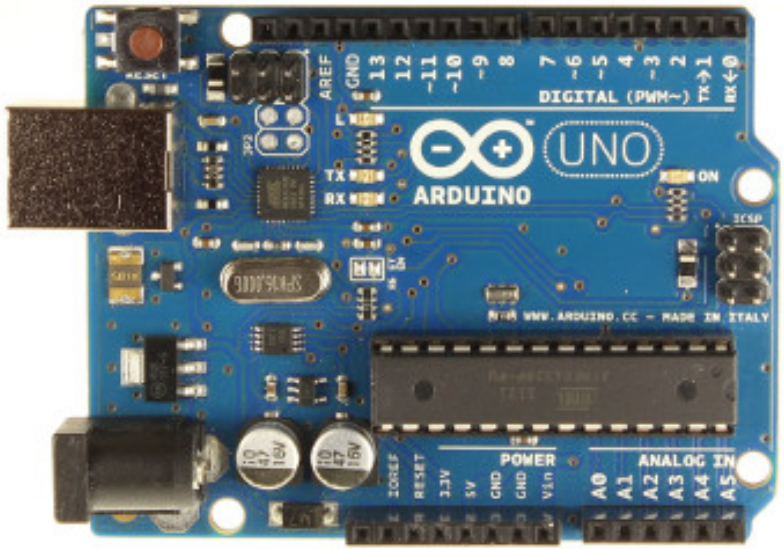
cs.karelia.ru

Microcontroller



- AVR, ARM, PIC and other
- 8/16/32 bit
- 1 MHz – 200 MHz
- different set of peripherals

Arduino

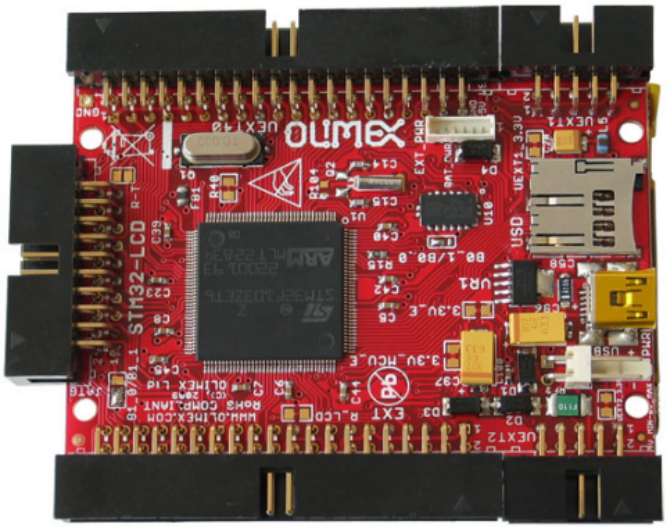


open-source electronics prototyping platform

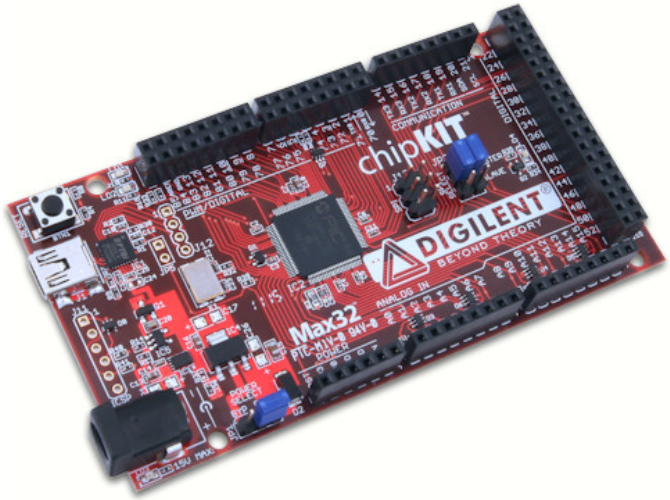
- quick start
- easy to use
- a large community

cs.karelia.ru

Other platforms



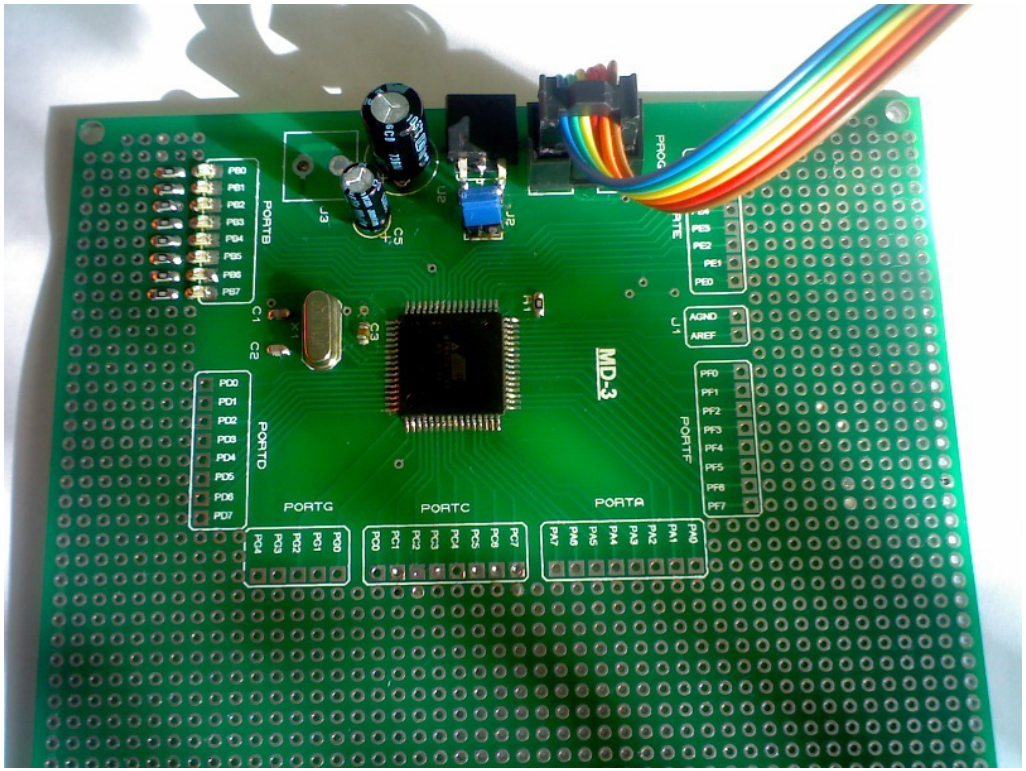
Cortex-M3



PIC32

MSP430

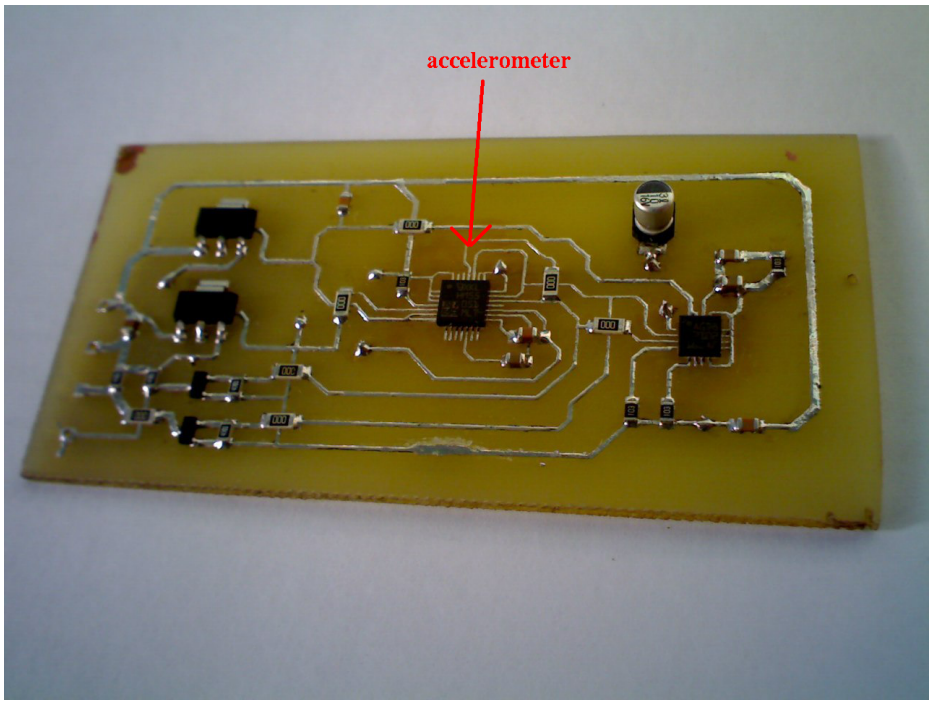
My board



- AVR 8-bit
- Atmega128
- 16 MHz

Accelerometer

- 3-axis accelerometer and 3-axis magnetometer
- I2C serial interface

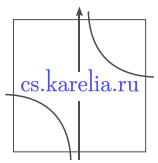


cs.karelia.ru

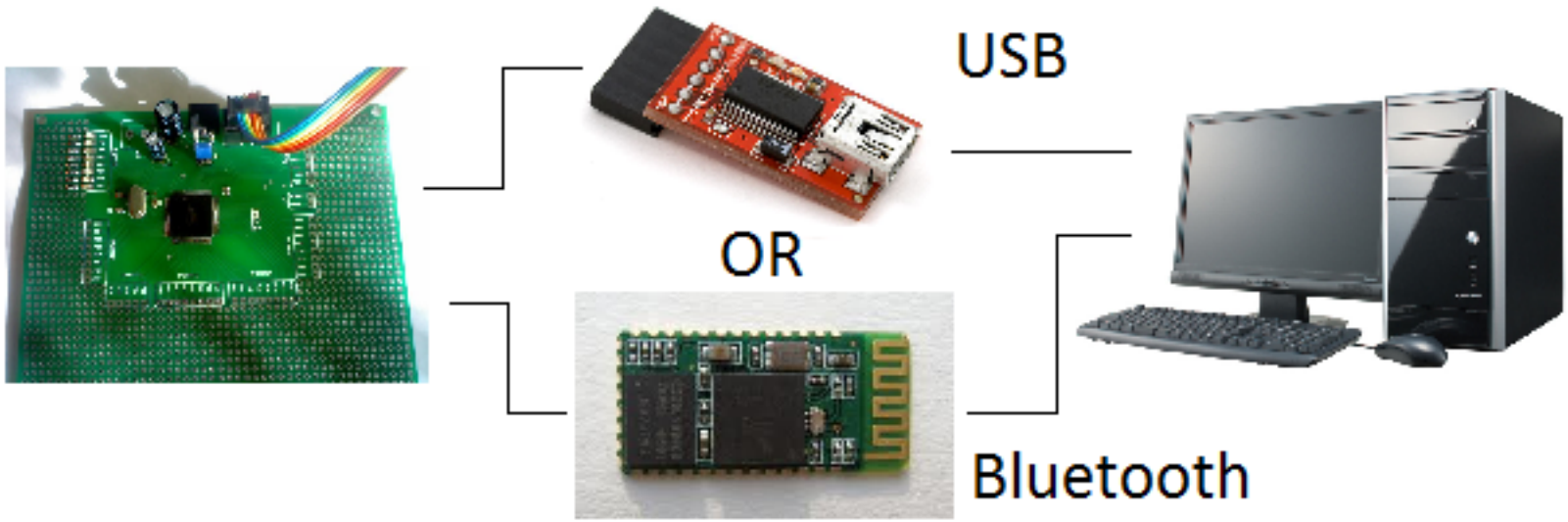
Sample code

```
}

// Writes an accelerometer register
void LSM303writeAccReg(uint8_t reg, uint8_t value)
{
    I2CWriteByte(ACC_ADDRESS_SA0_A_LOW, reg, value);
}

// Reads an accelerometer register
uint8_t LSM303readAccReg(uint8_t reg)
{
    uint8_t value;

    value = I2CReadByte(ACC_ADDRESS_SA0_A_LOW, reg);

    return value;
}

// Writes a magnetometer register
void LSM303writeMagReg(uint8_t reg, uint8_t value)
{
    I2CWriteByte(MAG_ADDRESS, reg, value);
}

// Reads a magnetometer register
uint8_t LSM303readMagReg(uint8_t reg)
{
    uint8_t value;
```

```
 * Created: 09.10.2011 14:58:51
 *  Author: Evgeny Tsvetkov
 */

#include "i2c.h"
#include "uart0.h"
#include "LSM303DLH.h"
#include "L3G4200D.h"
#include <stdio.h>

#define min(a, b) ((a)<(b) ? (a) : (b))
#define max(a, b) ((a)>(b) ? (a) : (b))

void Init(void)
{
    I2CInit();
    _delay_ms(50);
    Init_UART0();
    _delay_ms(50);
    LSM303enableDefault();
    _delay_ms(50);
    GYRenableDefault();
}

int main(void)
{
    int heading = 0;
    int pitch = 0;
    int roll = 0;
```



Connect to your computer



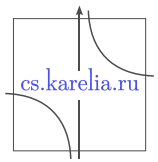
Data analysis

For quick results, i use:

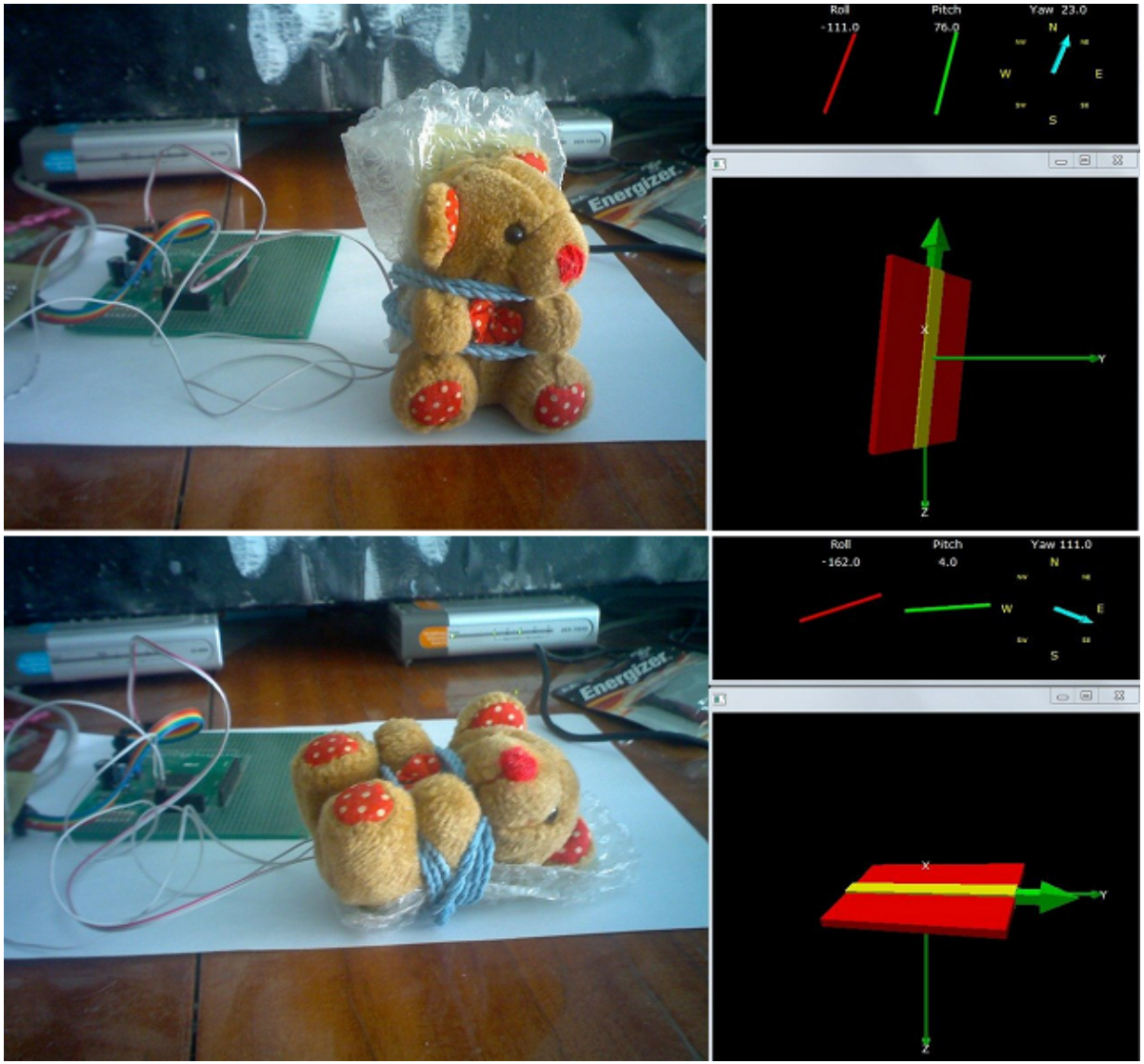
-  Python Scripts
-  Processing
-  LabVIEW

Also recommend:

-  Qt
-  C Sharp



Demonstration of the device



Conclusion

In the future:

- accumulation of data(simulation of falls)
- develop an algorithm which determines the fall
- introduce it in another project

This work is part of the project
"The Development Of Mobile Cardiac Monitoring System".

Thank you for your attention