

# DT-AVR *Application Note*

## DT-AVR

### AN225 – “Running Away” Game Controlled With Accelerometer And Gyroscope

Oleh: Tim IE

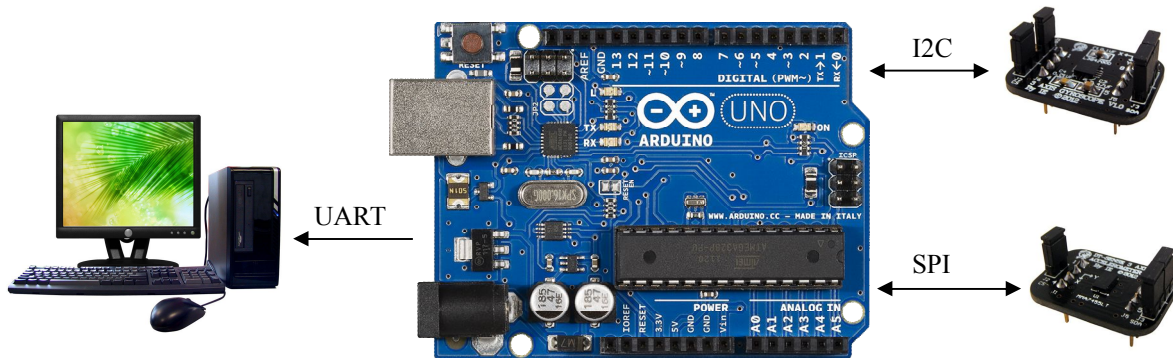
Perangkat elektronika telah banyak berkembang dan dilengkapi dengan berbagai *sensor*. Salah satu mamfaat dari *sensor* tersebut adalah mendeteksi setiap gerakan perangkat. *Sensor* yang digunakan untuk mendeteksi gerakan tersebut adalah *sensor accelerometer* dan *gyroscope*. *Accelerometer* sering digunakan untuk mengukur tingkat kemiringan benda dan *gyroscope* sering digunakan untuk mengukur kecepatan sudut benda. Kedua *sensor* ini sering digunakan pada perangkat *smartphone*, *game console*, dan *notebook*.

Pada aplikasi ini akan dibahas mengenai penggunaan modul *accelerometer* dan modul *gyroscope*. Modul *accelerometer* yang digunakan adalah DT-Sense 3 Axis Accelerometer dan modul *gyroscope* yang digunakan adalah DT-Sense 3 Axis Gyroscope. Sedangkan untuk kontroler untuk kedua modul tersebut menggunakan Arduino Uno. Aplikasi membutuhkan beberapa perangkat sebagai berikut:

- 1x DT-Sense 3 Axis Accelerometer
- 1x DT-Sense 3 Axis Gyroscope
- 1x Arduino Uno

Penggunaan modul *accelerometer* menggunakan komunikasi SPI dengan tegangan kerja sebesar 3,3 volt. Sedangkan penggunaan modul *gyroscope* menggunakan komunikasi I<sup>2</sup>C pada tegangan kerja 3,3 volt. Pada perangkat Arduino Uno telah disediakan supply tegangan 3,3 volt sehingga komunikasi antara perangkat modul sensor dengan perangkat kontroler dapat langsung dikoneksikan. Setelah data dari modul sensor diterima maka data dikirim oleh Arduino menuju komputer menggunakan komunikasi UART.

Adapun blok diagram dari aplikasi ini adalah sebagai berikut:



Gambar 1  
Blok Diagram AN225

**H**ubungan antar modul adalah sebagai berikut:

Arduino Uno	DT-Sense 3 Axis Accelerometer *
3.3V	Pin 8 (3V3D)
GND	Pin 7 (GND)
Pin 13 (SCK)	Pin 6 (SCL)
Pin 12 (MISO)	Pin 3 (SDO)
Pin 11 (MOSI)	Pin 5 (SDI)
Pin 10 (SS)	Pin 4 (CS)

\*) penomoran dan peletakan pin terdapat pada manual DT-Sense 3 Axis Accelerometer

**Tabel 1**  
**Hubungan Arduino Uno dengan DT-Sense 3 Axis Accelerometer**

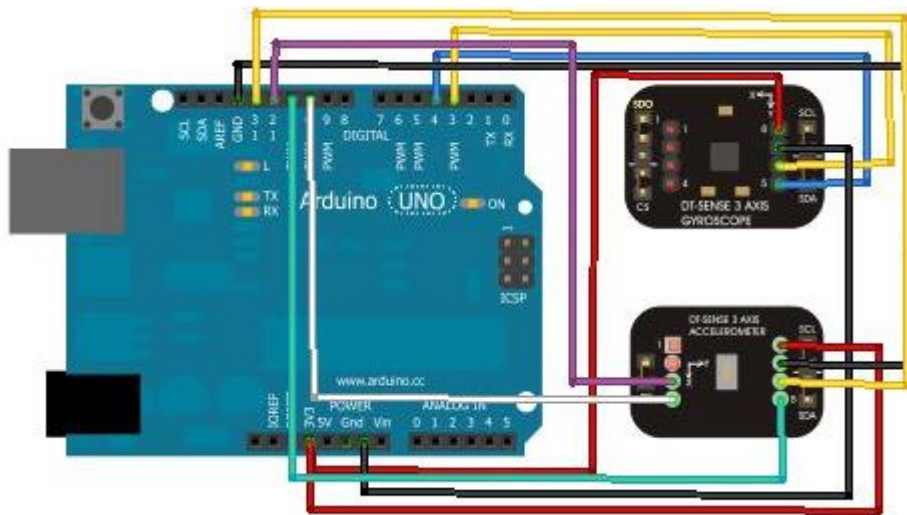
Arduino Uno	DT-Sense 3 Axis Gyroscope *
3.3V	Pin 8 (3V3D)
GND	Pin 7 (GND)
Pin 3	Pin 6 (SCL)
Pin 4	Pin 5 (SDA)

\*) penomoran dan peletakan pin terdapat pada manual DT-Sense 3 Axis Gyroscope

**Tabel 2**  
**Hubungan Arduino Uno dengan DT-Sense 3 Axis Gyroscope**

Tujuan dari aplikasi ini adalah untuk melakukan pembacaan terhadap sensor *gyroscope* dan *accelerometer*. Pembacaan modul sensor dengan menggunakan komunikasi serial. Komunikasi serial terdiri dari 3 macam yaitu komunikasi I<sup>2</sup>C, Komunikasi SPI, dan komunikasi UART. Pada aplikasi ini akan dijelaskan mengenai kegunaan dari ketiga jenis komunikasi tersebut terhadap masing – masing *sensor* (Untuk komunikasi UART lebih dijelaskan pada komunikasi antara perangkat Arduino Uno terhadap komputer).

Agar kedua modul sensor dapat melakukan komunikasi data dengan Arduino Uno maka dilakukan pengaturan pin J2 pada modul sensor. Apabila *jumper* dipasang pada pin J2, maka modul sensor akan menggunakan komunikasi I<sup>2</sup>C. Apabila pin J2 tidak menggunakan *jumper*, maka modul sensor akan menggunakan komunikasi SPI. Pada aplikasi ini komunikasi modul DT-Sense 3 Axis Gyroscope menggunakan komunikasi I<sup>2</sup>C dan modul DT-Sense 3 Axis Accelerometer menggunakan komunikasi SPI. Keterangan lebih lanjut mengenai *jumper* modul dapat dilihat pada *datasheet* masing-masing modul.



**Gambar 2**

**Ilustrasi Koneksi antar Modul DT-Sense 3 Axis Gyroscope, DT-Sense 3 Axis Accelerometer, dan Arduino Uno**

Modul Arduino Uno sangat mudah untuk dikoneksikan dengan modul DT-Sense 3 Axis Gyroscope dan DT-Sense 3 Axis Accelerometer. Ilustrasi koneksi antara modul dengan Arduino Uno dapat dilihat pada **Gambar 2**. Untuk koneksi DT-Sense 3 Axis Gyroscope dengan Arduino Uno hanya membutuhkan 4 buah kabel untuk komunikasi I<sup>2</sup>C. 4 buah kabel dihubungkan pada Arduino Uno pada pin 4 dan 3 untuk komunikasi I<sup>2</sup>C serta pin VCC dan GND untuk *supply* tegangan modul DT-Sense 3 Axis Gyroscope. Untuk koneksi DT-Sense 3 Axis Accelerometer membutuhkan 6 buah kabel untuk komunikasi SPI. 6 buah kabel dihubungkan pada Arduino Uno pada pin 10, 11, 12 dan 13 untuk komunikasi SPI serta pin VCC dan GND untuk *supply* tegangan modul DT-Sense 3 Axis Accelerometer.

Pada modul DT-Sense 3 Axis Gyroscope diperlukan pengaturan *jumper* J4 dan J3 untuk memberikan resistor *pull-up* pada jalur SDA dan SCL modul DT-Sense 3 Axis Gyroscope. Pengaturan *jumper* J4 dan J3 dapat dilihat pada Manual DT-Sense 3 Axis Gyroscope.

Setelah semua modul terhubung, periksa kembali semua koneksi pin. Pastikan koneksi VCC dan GND tidak terhubung singkat. Apabila semua koneksi sudah dipastikan benar, hubungkan kabel USB dari PC ke Arduino Uno, selanjutnya pastikan LED power pada Arduino Uno menyala dan PC sudah mengenali modul Arduino Uno.

Tambahkan *library* **Accel2Spi**, **Gyrol2C**, **SettingAccelI2C**, dan **SettingGyroSPI** pada *folder* "Libraries" di *folder* instalasi Arduino untuk menjalankan perintah I<sup>2</sup>C atau SPI pada Arduino IDE. *Library* dapat diakses atau ditambahkan pada program dengan memilih menu **Sketch – Import Library** pada Arduino IDE.

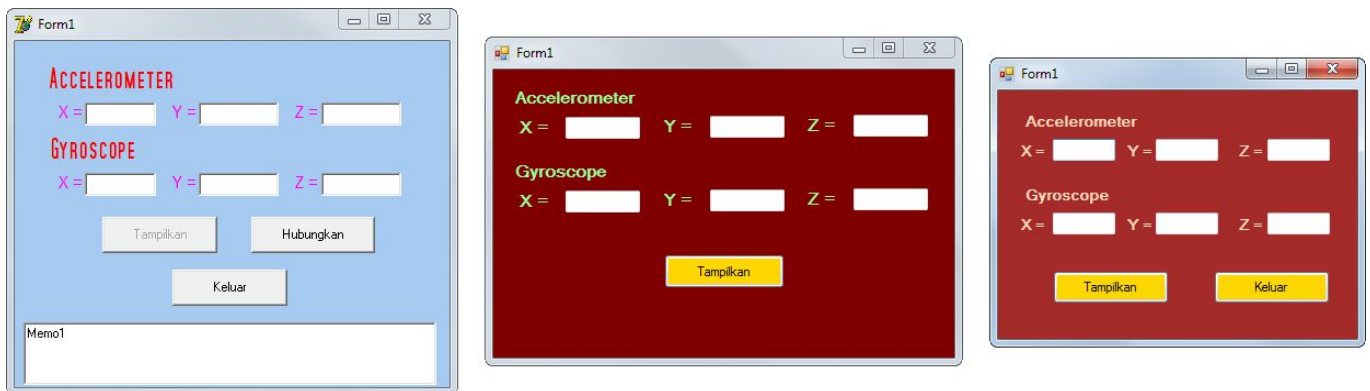
Modul	Library	Antarmuka
DT-Sense 3 Axis Accelerometer	<b>Accel2Spi</b>	SPI
	<b>SettingAccelI2C</b>	I <sup>2</sup> C
DT-Sense 3 Axis Gyroscope	<b>SettingGyroSPI</b>	SPI
	<b>Gyrol2C</b>	I <sup>2</sup> C

**Tabel 3**

**Hubungan Modul, Library, dan Antarmuka**

Buka *file* "project1\_AccelGyro\_library.ino" pada *folder* "project1\_AccelGyro\_library" menggunakan program Arduino IDE. Lakukan *upload sketch* ke Arduino Uno dan tunggu beberapa saat. Apabila instalasi *driver* sudah benar dan PC dapat mengenali Arduino Uno, maka program akan di-*upload* secara otomatis. Program akan membaca data sensor dan mengirimkan ke PC.

Tersedia contoh dan *listing* program untuk menampilkan data sensor pada PC yang dibuat menggunakan Delphi 7© dan Visual Studio 2010© (Visual Basic dan C#).



**Gambar 3**  
Tampilan Program Penampil Data Sensor pada PC (Delphi, VB, dan C#)

Pada program Delphi, tekan tombol **Hubungkan** untuk memilih COM *port* yang akan digunakan. Pada program Visual Studio, COM *port* diprogram ke COM40. Pastikan/Ubah melalui *Device Manager* bahwa Arduino telah dikenali menjadi COM40. Tekan tombol **Tampilkan** untuk membuka hubungan ke Arduino dan mulai membaca dan menampilkan data sensor. Tombol **Keluar** pada prinsipnya sama dengan tombol silang di ujung jendela, digunakan untuk keluar dari program.

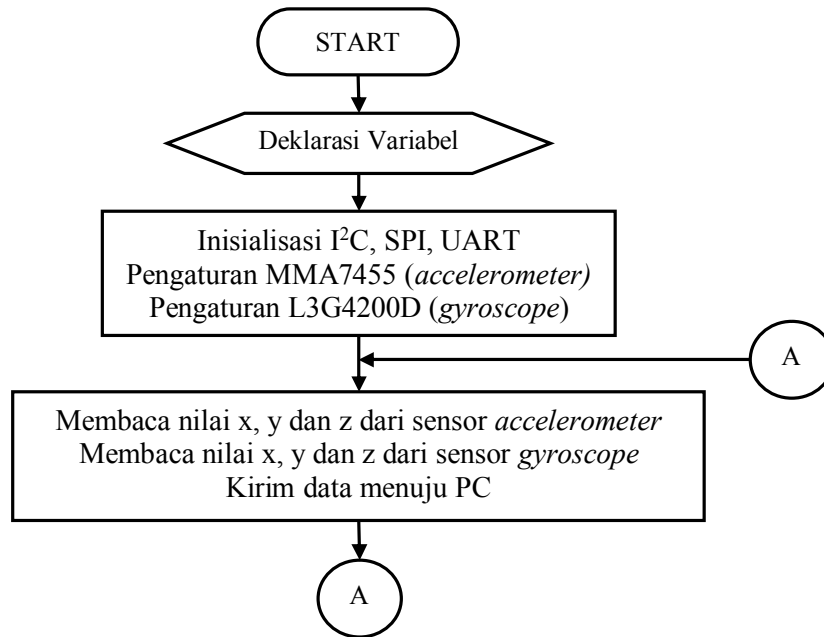
Tersedia juga contoh dan *listing* program untuk *game* sederhana yang dibuat menggunakan Visual Studio 2010© (C#). Berikut ini adalah tampilan *game* pada PC:



**Gambar 4**  
Tampilan *Game* pada PC

Untuk menjalankannya, pilih sensor yang akan digunakan di bagian kiri atas, **Gyroscope** atau **Accelerometer**. Lalu pilih kecepatan gerak musuh melalui pilihan **Speed** di bagian kanan bawah. Setelah itu tekan **Play**. Gerakkan sensor untuk menggerakkan tokoh anak kecil bertopi dalam menghindari kejaran musuh.

**F**lowchart dari program project1\_AccelGyro\_library.ino adalah sebagai berikut:



**Gambar 5**  
**Flowchart project1\_AccelGyro\_library.ino**

Penjelasan singkat dari alur program project1\_AccelGyro\_library.ino sebagai berikut :

1. Program proses inialisasi dan pengaturan sensor MMA7455 (*accelerometer*) dan L3G4200D (*gyroscope*) sebagai berikut:

```

void setup(void)
{
  // set up SPI
  //spiAccel_init();
  gyroSpi_init();
  // set up I2C
  //gyroi2c_init();
  I2CAccel_init();
  // set up UART
  Serial.begin(9600);
  delay(100);

  // Setting MMA7455 (accelerometer)
  setModeControl(0x05); // setting g-range menjadi 2g
  modeControl = readRegisterValueByteAccel(0x16);

  offsetX = SignAndValueToTwoComplementOffset('+',6);
  offsetY = SignAndValueToTwoComplementOffset('+',60);
  offsetZ = SignAndValueToTwoComplementOffset('-',15);

  setOffsetValueAccel(0x10,offsetX);
  setOffsetValueAccel(0x12,offsetY);
  setOffsetValueAccel(0x14,offsetZ);

  // Setting L3G4200D (gyroscope)
  //writeRegisterValueByteGyro(0xD0,0x20,0x0F); // Hanya untuk I2C
  writeRegisterValueByteGyro(0x20,0x0F); // Hanya untuk SPI
}
  
```

- Perintah `setModeControl` merupakan pengaturan *mode control register*. Pada aplikasi AN225, MMA7455 diatur pada pengukuran 2g dengan sensitivitas 64 LSB/g pada 8-bit *output* yang berarti pengukuran tiap 1-bit *output* sama dengan 15,625 mg. *Mode control register* diatur pada kondisi pengukuran. Berikut adalah program dari `setModeControl`:

```
void writeRegisterValueByteAccel(unsigned char address,unsigned char value) {
    digitalWrite( SLAVE_SELECT_PIN, LOW );
    byte command = SPI_WRITE | ( address << 1 );
    SPI.transfer( command );
    SPI.transfer( value );
    digitalWrite( SLAVE_SELECT_PIN, HIGH );
    delay( 100 );
}

void setModeControl(unsigned char value) {
    writeRegisterValueByteAccel(REG_MCTL,value);
}
```

- Perintah selanjutnya setelah mengatur *mode control register* adalah perintah pengaturan nilai *offset drift* dari MMA7455. Berikut ini adalah program dari pengaturan dari nilai *offset drift* dari sensor:

```
unsigned int SignAndValueToTwoComplementOffset(unsigned char sign,unsigned int value) {

    if (sign=='-') {
        value = ~value;
        value = value + 1;
        value = value & 0x7FF;
    }
    return value;
}

void setOffsetValueAccel(unsigned char address,unsigned int value) {
    digitalWrite( SLAVE_SELECT_PIN, LOW );
    byte command = SPI_WRITE | ( address << 1 );
    SPI.transfer( command );
    SPI.transfer( value%256 );
    SPI.transfer(value/256);
    digitalWrite( SLAVE_SELECT_PIN, HIGH );
    delay( 100 );
}
```

- Pengaturan L3G4200D adalah pengaturan *control register*. Pada pengaturan ini data pada sumbu x, y, dan z diaktifkan. Berikut ini adalah program dari pengaturan L3G4200D:

```
void writeRegisterValueByteGyro(unsigned char I2Caddress,unsigned char address,
unsigned char value)
{
    i2c_start(I2Caddress);
    i2c_write(address);
    i2c_write(value);
    i2c_stop();
}
```

5. Program proses pengambilan data dari *Sensor MMA7455 (accelerometer)* dan *L3G4200D (gyroscope)* sebagai berikut:

```
void loop(void)
{
    // Proses pengambilan data MMA7455 (accelerometer)
    x = readRegisterValueWordAccel(0x00);
    y = readRegisterValueWordAccel(0x02);
    z = readRegisterValueWordAccel(0x04);
    TenBitTwoComplementToSignAndValue(x,&signX,&valueX);
    TenBitTwoComplementToSignAndValue(y,&signY,&valueY);
    TenBitTwoComplementToSignAndValue(z,&signZ,&valueZ);
    //movingAverage(valueX, valueY, valueZ);
    average(valueX, valueY, valueZ);

    readL3G4200D(); // Proses pengambilan data gyroscope

    // Pengiriman Data accelerometer dan gyroscope menuju PC
    //Serial.write('a');
    //Serial.write(125);
    Serial.print('\n');
    //SerialGyro1();
    SerialGyro2();
    //SerialAccel1(signX, valueX, signY, valueY, signZ, valueZ);
    SerialAccel2(signX, valueX, signY, valueY, signZ, valueZ);
}

```

6. Proses pengambilan data dari MMA7455 dengan proses pengolahan data sebagai berikut:

```
unsigned int readRegisterValueWordAccel(unsigned char address) {
    unsigned int value,tempValue;

    digitalWrite( SLAVE_SELECT_PIN, LOW );
    byte command = SPI_READ | ( address << 1 );
    SPI.transfer( command );
    tempValue = SPI.transfer( 0x01 );
    value = SPI.transfer( 0x00 );
    digitalWrite( SLAVE_SELECT_PIN, HIGH );
    value = value << 8;
    value += tempValue;

    return value;
}

void TenBitTwoComplementToSignAndValue(unsigned int input,unsigned char* sign,
unsigned int* value) {
    if ((input & 0x0200)==0x0200) {
        *sign = '-';
        input = ~input;
        input = input & 0x01FF;
        input = input + 1;
        *value = input;
    }
    else {
        *sign = '+';
        *value = input;
    }
}

```

7. Proses pengambilan data dari L3G4200D dengan proses pengolahan data sebagai berikut:

```
unsigned char readRegisterValueByteGyro(unsigned char address, unsigned char ack) {
    unsigned char value;

    i2c_start(0xD0);
    i2c_write(address);
    i2c_stop();

    i2c_start(0xD1);
    value = i2c_read(ack);
    i2c_stop();

    return value;
}

void readL3G4200D(void) {

    a = readRegisterValueByteGyro(0x28,1);
    b = readRegisterValueByteGyro(0x29,1);
    c = readRegisterValueByteGyro(0x2A,1);
    d = readRegisterValueByteGyro(0x2B,1);
    e = readRegisterValueByteGyro(0x2C,1);
    f = readRegisterValueByteGyro(0x2D,0);

    x_gyro = (int)b;
    x_gyro = x_gyro << 8;
    x_gyro |= a;
    x_gyro = x_gyro /128;

    y_gyro = (int)d;
    y_gyro = y_gyro << 8;
    y_gyro |= c;
    y_gyro = y_gyro /128;

    z_gyro = (int)f;
    z_gyro = z_gyro << 8;
    z_gyro |= e;
    z_gyro = z_gyro /128;

}
```

Penjelasan mengenai *library Accel2Spi* sebagai berikut:

1. **setOffsetValueAccel(unsigned char address,unsigned int value)**  
Perintah ini digunakan untuk mengatur mode dari sensor Accelerometer supaya mendapatkan nilai *offset*. **Address** pada perintah ini digunakan sebagai alamat register. **Value** pada perintah ini digunakan untuk mengatur nilai offset pada register sensor Accelerometer.
2. **setModeControl(unsigned char value)**  
Perintah ini digunakan untuk mengatur *g-range* pada sensor Accelerometer. **Value** pada perintah ini digunakan untuk mengatur nilai *g-range*.
3. **readRegisterValueByteAccel(unsigned char address)**  
Perintah ini digunakan untuk melakukan pembacaan mode pada sensor Accelerometer. **Address** pada perintah ini menyatakan alamat register yang akan dibaca.
4. **readRegisterValueWordAccel(unsigned char address)**  
Perintah ini digunakan untuk melakukan pembacaan nilai hasil pengukuran yang dilakukan oleh sensor Accelerometer. **Address** pada perintah ini menyatakan alamat register sumbu yaw/roll/pitch yang akan dibaca.



5. **TenBitTwoComplementToSignAndValue(unsigned int input,unsigned char\* sign, unsigned int\* value)**  
Perintah ini digunakan untuk memproses nilai hasil pengukuran sensor Accelerometer. Hasil proses tersebut merupakan nilai yang telah memiliki tanda positif atau negatif. **Input** pada perintah ini menyatakan nilai dari pengukuran sensor Accelerometer yang akan diproses. **Sign** pada perintah ini menyatakan tanda positif atau negatif. **Value** pada perintah ini merupakan *output*.
6. **SignAndValueToTwoComplementOffset(unsigned char sign, unsigned int value)**  
Perintah ini digunakan untuk mengatur nilai *offset* dari sensor Accelerometer. **Sign** pada perintah ini menyatakan tanda positif atau negatif. **Value** pada perintah ini merupakan nilai *offset* yang diinginkan.
7. **movingAverage(int dataX, int dataY, int dataZ)**  
Perintah ini digunakan untuk melakukan filter nilai sensor Accelerometer dengan metode Moving Average. **DataX**, **dataY**, dan **dataZ** merupakan *input* dari masing-masing sumbu pada sensor Accelerometer.
8. **average(int dataX, int dataY, int dataZ)**  
Perintah ini digunakan untuk melakukan filter nilai sensor Accelerometer dengan metode rata-rata. **DataX**, **dataY**, dan **dataZ** merupakan *input* dari masing-masing sumbu pada sensor Accelerometer.
9. **SerialAccel1(int signx, int accelx, int signy, int accely, int signz, int accelz)**  
Perintah ini digunakan untuk mengirim nilai dari sensor Accelerometer menuju PC yang akan ditampilkan pada program Visual Studio. **Signx**, **signy**, dan **signz** merupakan *input* yang dari tanda positif atau negatif hasil pemrosesan. **Accelx**, **accely** dan **accelz** merupakan *input* nilai dari sensor Accelerometer.
10. **SerialAccel2(int signx, int accelx, int signy, int accely, int signz, int accelz)**  
Perintah ini digunakan untuk mengirim nilai dari sensor Accelerometer menuju PC yang akan ditampilkan pada program Delphi. **Signx**, **signy**, dan **signz** merupakan *input* yang dari tanda positif atau negatif hasil pemrosesan. **Accelx**, **accely** dan **accelz** merupakan *input* nilai dari sensor Accelerometer.
11. **spiAccel\_init()**  
Perintah ini digunakan untuk mengaktifkan komunikasi SPI pada sensor Accelerometer.

Penjelasan mengenai *library GyroI2C* sebagai berikut:

1. **writeRegisterValueByteGyro(unsigned char I2Caddress, unsigned char address, unsigned char value)**  
Perintah ini digunakan untuk melakukan pengaturan mode pada sensor Gyroscope. **I2Caddress** menyatakan alamat perangkat sensor Gyroscope. **Address** pada perintah ini menyatakan alamat register sensor Gyroscope. **Value** pada perintah ini digunakan untuk mengatur nilai pada register *control*.
2. **readRegisterValueByteGyro(unsigned char address, unsigned char ack)**  
Perintah ini digunakan untuk melakukan pembacaan nilai sensor Gyroscope pada 1 sumbu. Pemilihan tiap sumbu dapat diatur sesuai keinginan. **Address** pada perintah ini menyatakan alamat register sensor Gyroscope. **Ack** pada perintah ini merupakan nilai *acknowledge*.
3. **readL3G4200D(void)**  
Perintah ini digunakan untuk melakukan pembacaan nilai sensor Gyroscope pada 3 sumbu (*yaw*, *roll*, dan *pitch*).
4. **SerialGyro1()**  
Perintah ini digunakan untuk melakukan pengiriman nilai sensor Gyroscope menuju PC yang akan ditampilkan pada program Visual Studio.
5. **SerialGyro2()**  
Perintah ini digunakan untuk melakukan pengiriman nilai sensor Gyroscope menuju PC yang akan ditampilkan pada program Delphi.
6. **gyroi2c\_init()**  
Perintah ini digunakan untuk mengaktifkan komunikasi I<sup>2</sup>C pada sensor Gyroscope.

Penjelasan mengenai *library SettingAccelI2C* sebagai berikut:

1. Fungsi-fungsi yang digunakan pada *library* ini tidak jauh berbeda dengan *library Accel2Spi*. Perbedaan pada *library* ini terletak pada proses pengaturan komunikasi antar perangkat. Fungsi yang digunakan untuk mengatur komunikasi antar perangkat adalah **I2CAccel\_init()**. Perintah **I2CAccel\_init()** merupakan perintah untuk mengatur komunikasi sensor Accelerometer menjadi I<sup>2</sup>C.

*Library SettingGyroSPI* tidak jauh berbeda dengan *library GyroI2C*. Penjelasan perbedaan *library SettingGyroSPI* sebagai berikut:

1. **writeRegisterValueByteGyro(unsigned char address, unsigned char value)**  
Perintah ini digunakan untuk melakukan pengaturan mode pada sensor Gyroscope. Pada perintah ini tidak menggunakan **I2Caddress**.
2. **gyroSpi\_init()**  
Perintah ini digunakan untuk mengaktifkan komunikasi SPI pada sensor Gyroscope.

**L**isting program aplikasi ini terdapat pada **AN225.ZIP**.

**S**elamat berinovasi!

*All trademarks, company names, product names and trade names are the property of their respective owners.  
All softwares are copyright by their respective creators and/or software publishers.*