

LED Example

```
int main(void)
{
    u8 i = 0;

    System_Configuration();

    while(1)
    {
        GPIO_WriteBit(GPIOC, GPIO_Pin_13, i);           // LED_TX
        GPIO_WriteBit(GPIOC, GPIO_Pin_14, !i);          // LED_RX
        GPIO_WriteBit(GPIOC, GPIO_Pin_15, i);           // PLAY
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, !i);          // EDIT
        GPIO_WriteBit(GPIOC, GPIO_Pin_6, i);            // MNG

        i = !i;

        mDelay(150);

    };
}
```

Results

Control LED's blink

Button Example

```
int main(void)
{
    u8 sw_mode, sw_start;

    System_Configuration();

    while(1)
    {
        sw_mode = GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_14);
        sw_start = GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_15);

        GPIO_WriteBit(GPIOC, GPIO_Pin_15, sw_mode);
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, sw_start);

    };
}
```

Results

Press the MODE button, the PLAY LED (red) and EDIT LED (green) turn on.

Serial Communications Example

```
int main(void)
{
    u16 recv;
    u8 led = 0;

    System_Configuration();

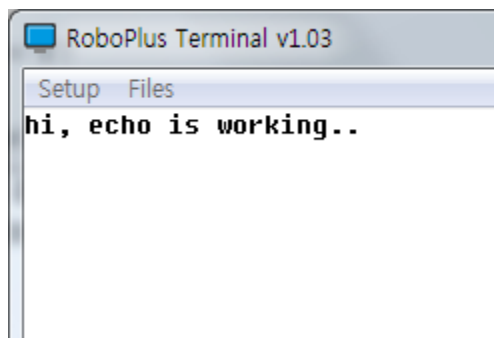
    USART_Configuration(USART_PC, 57600);
    USART_ITConfig(USART3, USART_IT_RXNE, DISABLE);

    while(1)
    {
        if(USART_GetFlagStatus(USART3, USART_FLAG_RXNE) != 0)
        {
            GPIO_WriteBit(GPIOC, GPIO_Pin_14, led);
            led = !led;
            recv = USART_ReceiveData(USART3);
            USART_SendData(USART3, recv);
            USART_ClearFlag(USART3, USART_FLAG_RXNE);
        }

    };
}
```

Results

The terminal window outputs an echo after input a value is entered. The RX LED toggles.



Buzzer Example – GPIO

```
int main(void)
{
    u8 i;

    System_Configuration();

    while(1)
    {
        if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_14) == 0)
        {
            GPIO_WriteBit(GPIOB, GPIO_Pin_9, i);
            mDelay(1);
        }

        i = !i;
    };
}
```

Results

Press the MODE button and the buzzer sounds off.

Dynamixel Power Control Example

```
int main(void)
{
    System_Configuration();

    while(1)
    {
        GPIO_WriteBit(GPIOB, GPIO_Pin_8, 1);
        mDelay(500);
        GPIO_WriteBit(GPIOB, GPIO_Pin_8, 0);
        mDelay(500);

    };
}
```

Results

Dynamixel's LED blinks at 0.5 seconds intervals (note that Dynamixel has to be connected to the CM-730).

Dynamixel read/write Example

```
int main(void)
{
    int bMovin, index, status;
    int goalPos[2] = {0, 1023};

    System_Configuration();

    GPIO_WriteBit(GPIOB, GPIO_Pin_8, 1);    // DXL Power On

    while(1)
    {
        bMovin = dxl_read_byte(1, 46);
        status = dxl_get_result();
        GPIO_WriteBit(GPIOC, GPIO_Pin_15, bMovin);    // PLAY

        if ((bMovin == 0) && (status == 1))
        {
            index = !index;
            dxl_write_word(1, 30, goalPos[index]);
        }

    };
}
```

Results

Dynamixel's PLAY LED blinks momentarily after reaching goal/target position.

Accelerometer Read Example

```
void ACC_Setting(void)
{
    u16 i;
    u16 data;

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);

    GPIO_ResetBits(PORT_SIG_ACC_CS, PIN_SIG_ACC_CS);

    i=0x2027;
    SPI_I2S_SendData(SPI2, i);

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);
    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET);

    data = SPI_I2S_ReceiveData(SPI2);

    GPIO_SetBits(PORT_SIG_ACC_CS, PIN_SIG_ACC_CS);
}

u8 getACC(u8 address)
{
    u16 data;
    u16 add;

    add = address << 8 ;
    add |= 0x8000;

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);

    GPIO_ResetBits(PORT_SIG_ACC_CS, PIN_SIG_ACC_CS);

    SPI_I2S_SendData(SPI2, add);

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);
    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET);

    data = SPI_I2S_ReceiveData(SPI2);

    GPIO_SetBits(PORT_SIG_ACC_CS, PIN_SIG_ACC_CS);

    return (u8)(data&0x00FF);
}

void GetAccXYZ(u16 *x, u16 *y, u16 *z)
{
    *x = (((u16)getACC(0x28+1))<<8)+getACC(0x28);
    *y = (((u16)getACC(0x28+3))<<8)+getACC(0x28+2);
    *z = (((u16)getACC(0x28+5))<<8)+getACC(0x28+4);
}
```

```

int main(void)
{
    u16 x, y, z;

    System_Configuration();

    ACC_Setting();

    USART_Configuration(USART_PC, 57600);

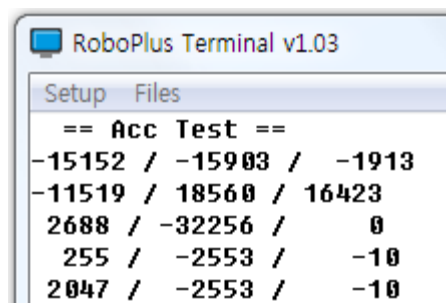
    TxDString(USART_PC, " == Acc Test == \r\n");

    while(1)
    {
        GetAccXYZ(&x, &y, &z);
        TxD_Dec_S16((s16)x);
        TxDString(USART_PC, " / ");
        TxD_Dec_S16((s16)y);
        TxDString(USART_PC, " / ");
        TxD_Dec_S16((s16)z);
        TxDString(USART_PC, "\r\n");
        mDelay(100);

    };
}

```

Results



```

RoboPlus Terminal v1.03
Setup Files
== Acc Test ==
-15152 / -15903 / -1913
-11519 / 18560 / 16423
2688 / -32256 / 0
255 / -2553 / -10
2047 / -2553 / -10

```


Gyroscope Read Example

```
void Gyro_Setting(void)
{
    u16 i;
    u16 data;

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);

    GPIO_ResetBits(PORT_SIG_GYRO_CS, PIN_SIG_GYRO_CS);
    i=0x200F;
    SPI_I2S_SendData(SPI2,i);

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);
    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET);

    data = SPI_I2S_ReceiveData(SPI2);

    GPIO_SetBits(PORT_SIG_GYRO_CS, PIN_SIG_GYRO_CS);
}

u8 getGyro(u8 address)
{
    u16 data;
    u16 add;

    add = address << 8 ;
    add |= 0x8000;

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);

    GPIO_ResetBits(PORT_SIG_GYRO_CS, PIN_SIG_GYRO_CS);

    SPI_I2S_SendData(SPI2,add);

    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);
    while (SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET);

    data = SPI_I2S_ReceiveData(SPI2);

    GPIO_SetBits(PORT_SIG_GYRO_CS, PIN_SIG_GYRO_CS);

    return (u8)(data&0x00FF);
}

void GetGyroXYZ(u16 *x, u16 *y, u16 *z)
{
    *x = (((u16)getGyro(0x28+1))<<8)+getGyro(0x28);
    *y = (((u16)getGyro(0x28+3))<<8)+getGyro(0x28+2);
    *z = (((u16)getGyro(0x28+5))<<8)+getGyro(0x28+4);
}
```

```
int main(void)
{
    u16 x, y, z;

    System_Configuration();

    Gyro_Setting();

    USART_Configuration(USART_PC, 57600);

    TxDString(USART_PC, " == Gyro Test == \r\n");

    while(1)
    {
        GetGyroXYZ(&x, &y, &z);
        TxD_Dec_S16((s16)x);
        TxDString(USART_PC, " / ");
        TxD_Dec_S16((s16)y);
        TxDString(USART_PC, " / ");
        TxD_Dec_S16((s16)z);
        TxDString(USART_PC, "\r\n");
        mDelay(100);

    };
}
```