Arduino Bluetoothシールド/ Arduino Bluetoothライブラリ/ Android Bluetoothライブラリ 説明書

Arduino Bluetoothシールド

使用Bluetoothモジュール:FB155BC

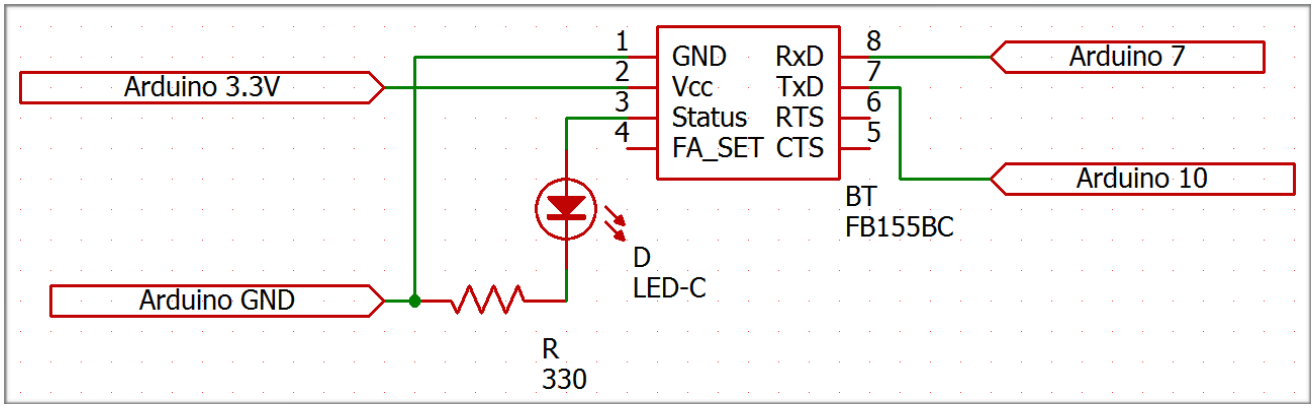

図 1 回路図

Arduino Bluetoothライブラリをそのまま使用するのであれば、図 1 の通りのピン配置で設計、制作する。ライブラリを改変する、または使用しないのであれば、ピン配置は適当に変えることもできる。

Arduino Bluetoothライブラリ

利用方法

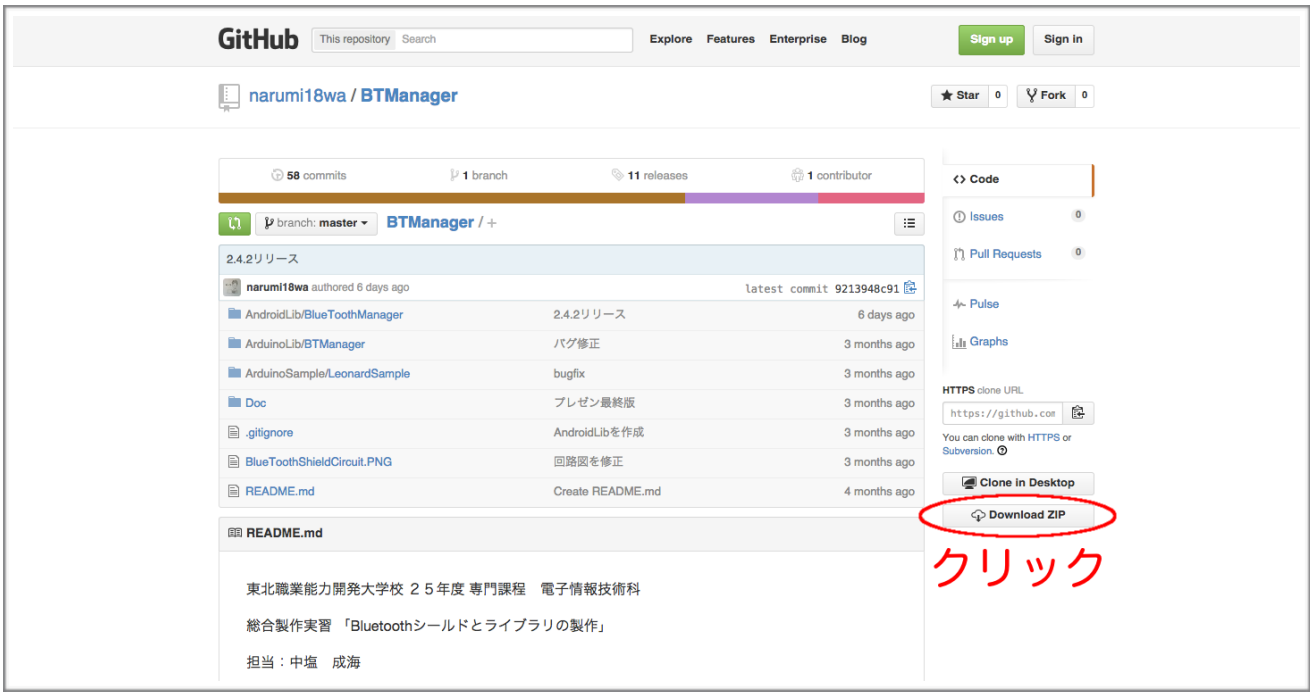

図２全てダウンロード

図２に従い、全てのファイルをZIPでダウンロードします。

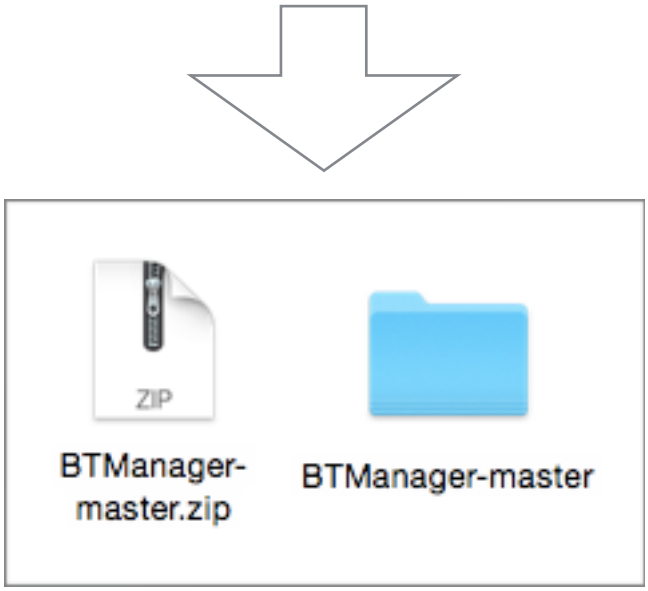

図３展開してください

図３のようにわかり易い場所に展開してください。

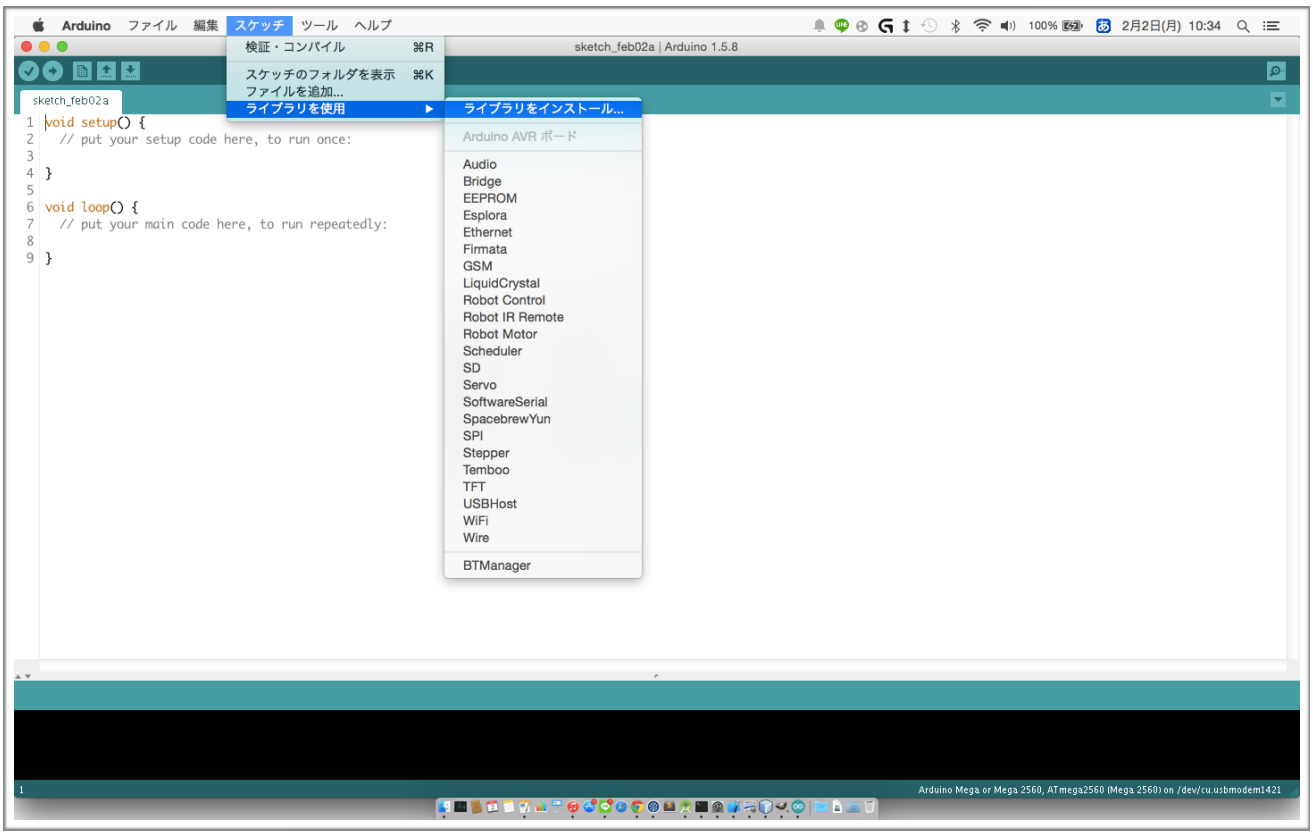

図４ライブラリをインストールをクリック

図４のように、ArduinoIDEを起動し、ツールバーからライブラリをインストール…をクリックしてください。

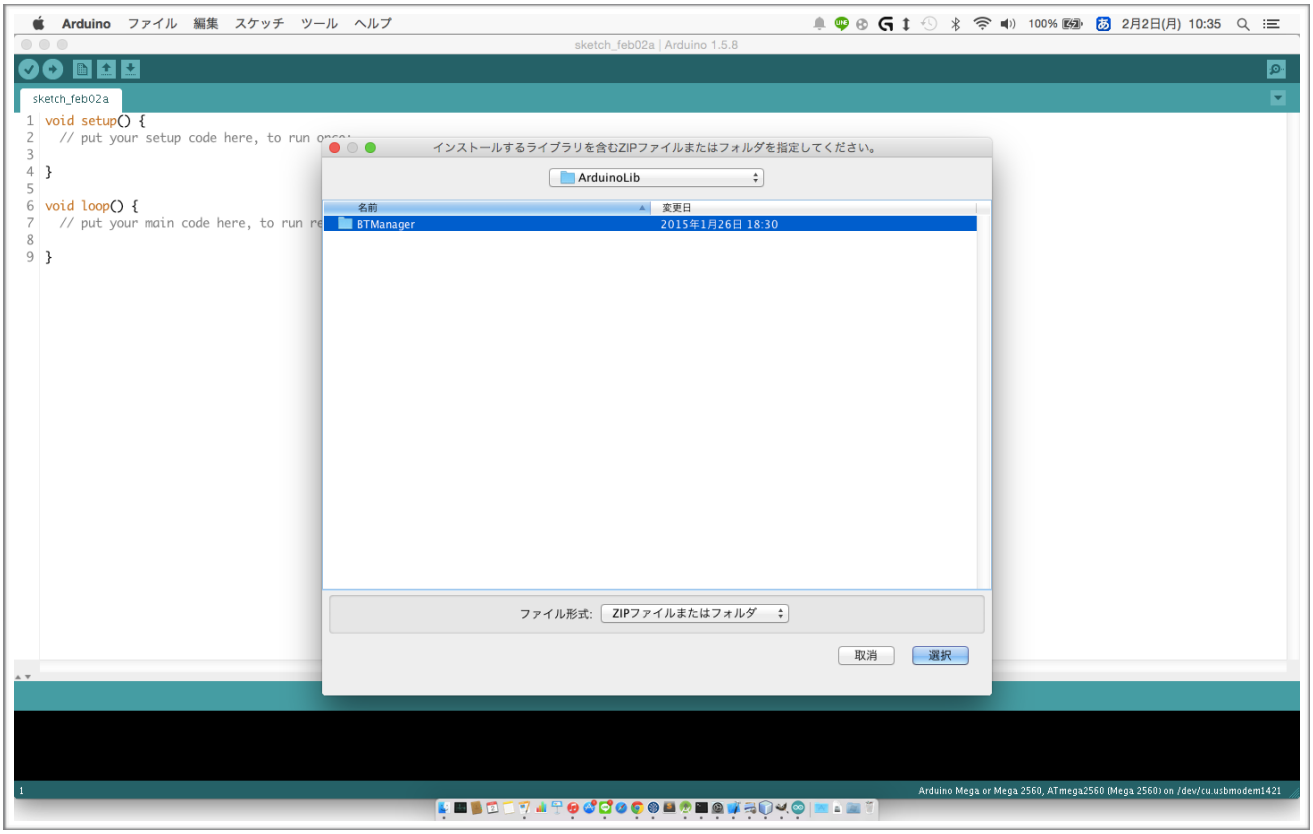

図５BTManagerを開く

図５のように展開したファイルの中の(展開したフォルダ)/BTManager-master/ArduinoLib/BTManagerを開く

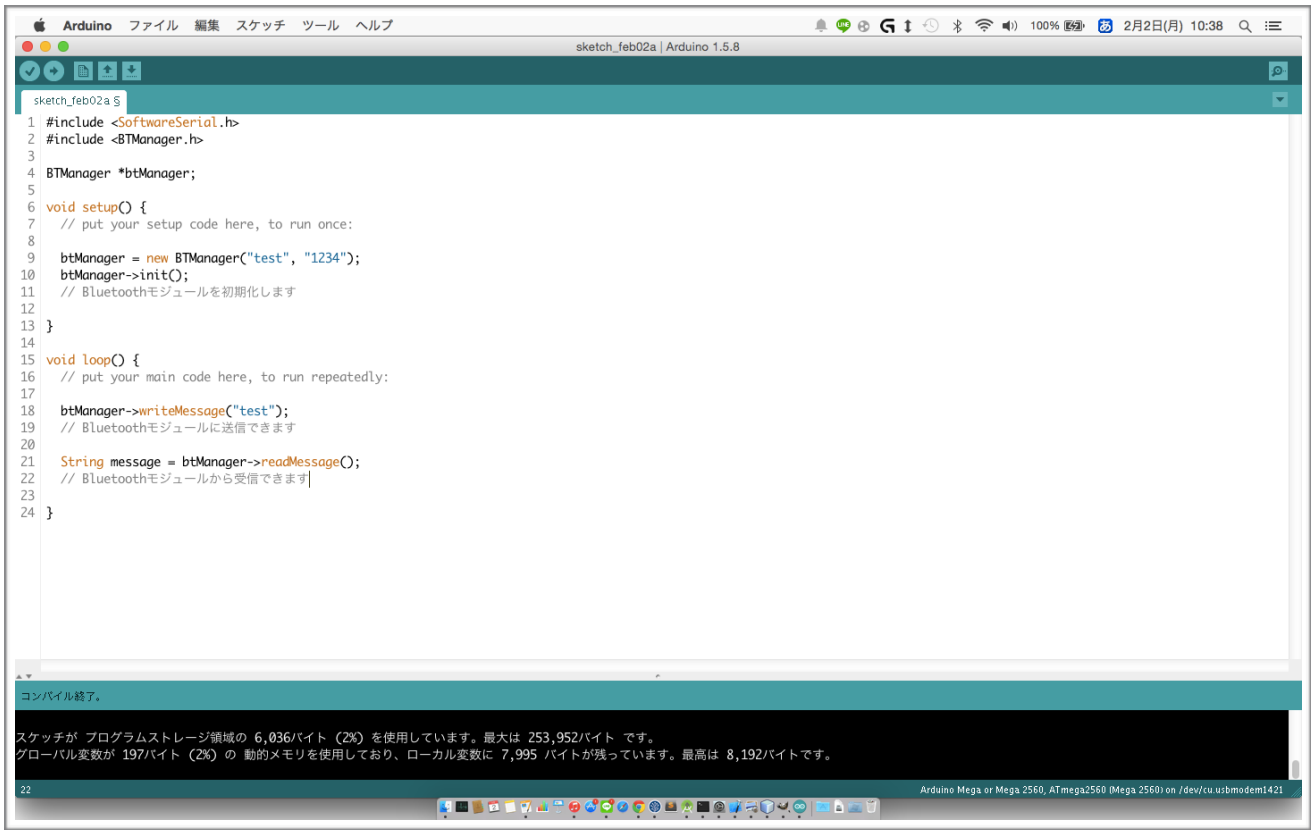

図６必要最低限のコード

図６を参考にBTManagerを動くように設定する。loop内の実装は例であって、好きに実装して良い。

仕様

Sketchからアクセスできるメソッドは以下の通りの仕様となっている

BTManager(String btName, String btPassWord);

コンストラクタ。btNameは”他の端末から見える名前”、btPassWordは”パスワード”をそれぞれ設定できる。

void init();

Bluetoothモジュールをコンストラクタの設定で初期化する。

String readMessage();

Bluetoothモジュールに送信されているメッセージを読む。

void writeMessage(String message);

Bluetoothモジュールからメッセージを送信する。

Android Bluetoothライブラリ

利用方法

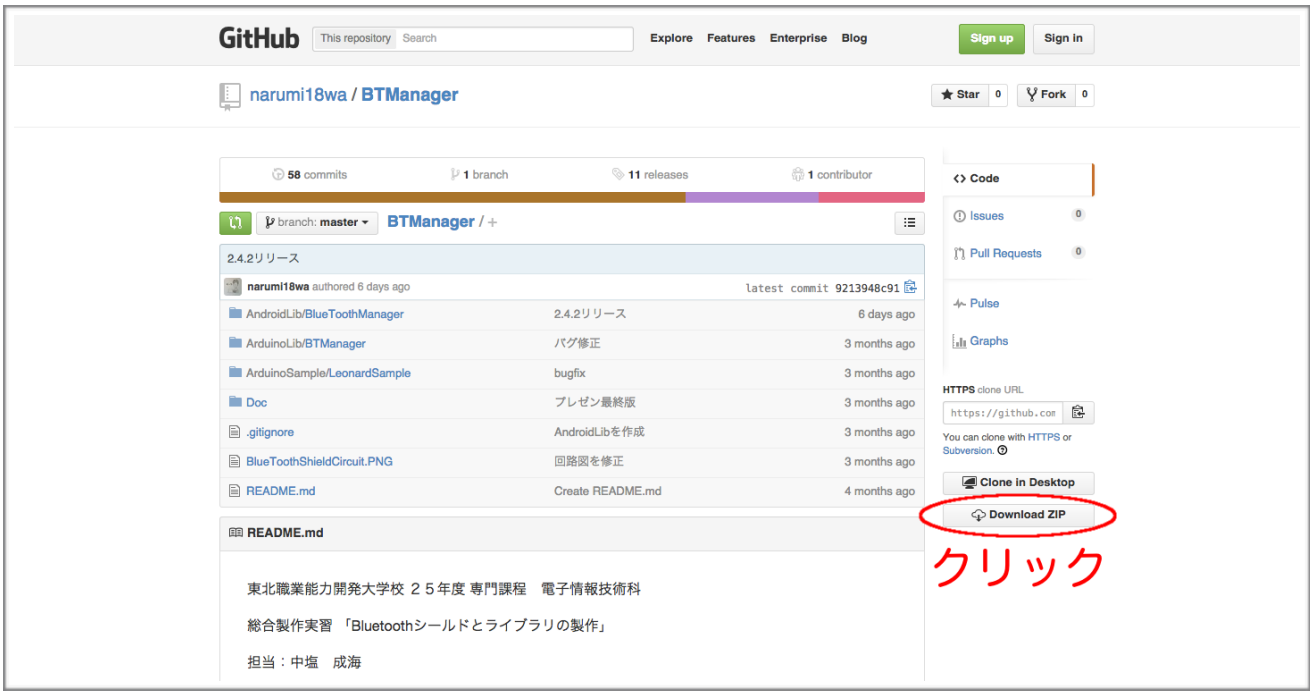

図７全てダウンロード

図７に従い、全てのファイルをZIPでダウンロードします。

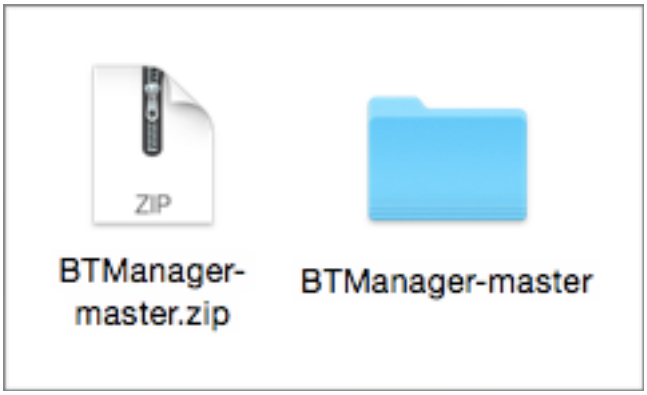

図８展開してください

図８のようにわかり易い場所に展開してください。

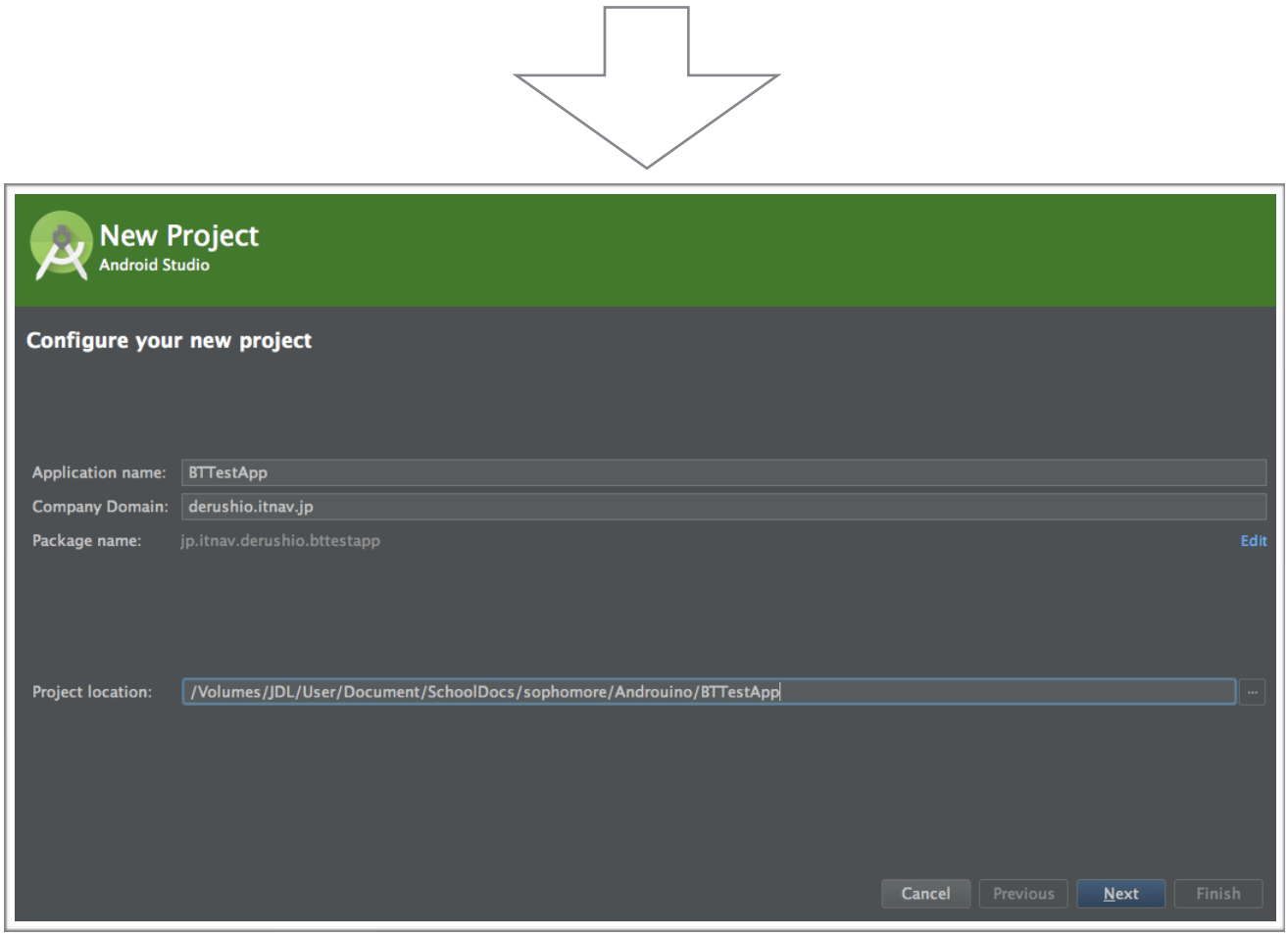

図９プロジェクトを作成してください

あなたの作りたいプロジェクトを作成してください。まず、テスト用のプロジェクトを作成しても良いかもしれません。

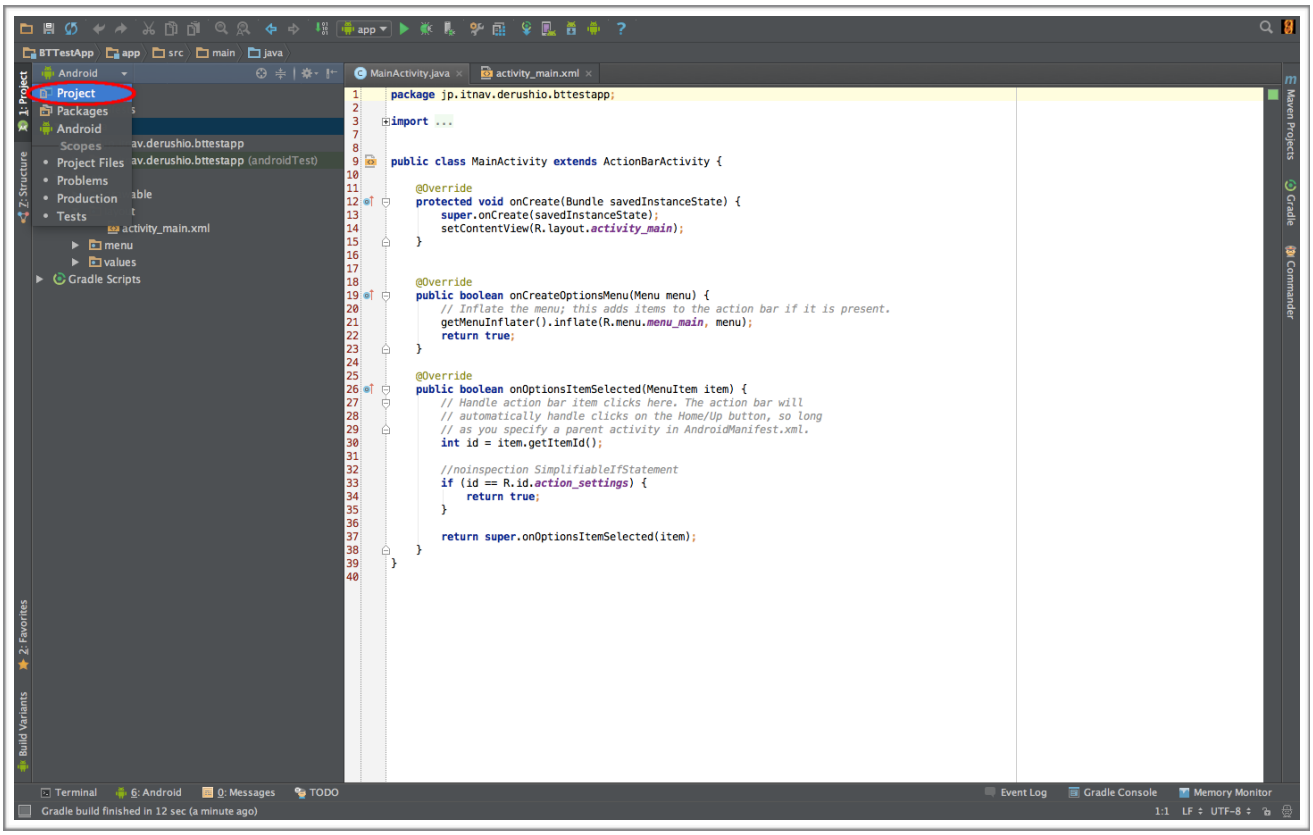

図１０フォルダツリーを表示

フォルダツリーを表示させるために、左下のメニュー（□マーク）から Projectを選び、プルダウンメニューから、Projectを選択します。すでに Projectになっている場合はそのままで大丈夫です。

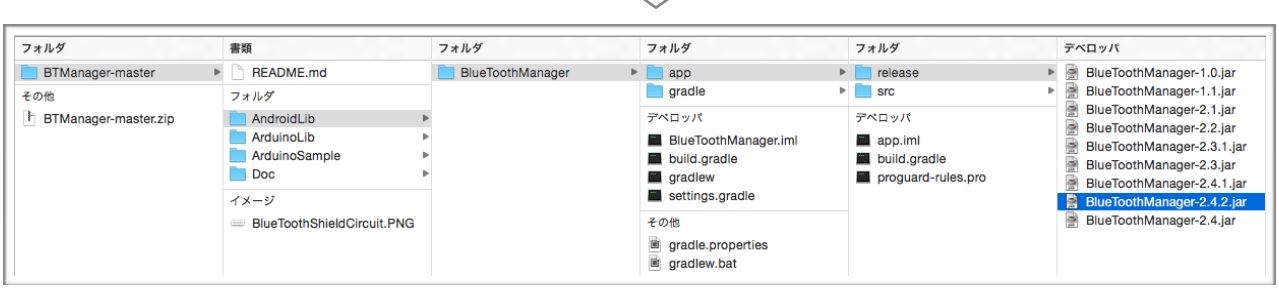

図１１jarファイルをコピー

図５のように(展開したフォルダ)/BTManager-master/AndroidLib/BlueToothManager/app/release/BlueToothManager-2.4.2.jar（最新版があればそちら）をコピーしてください。

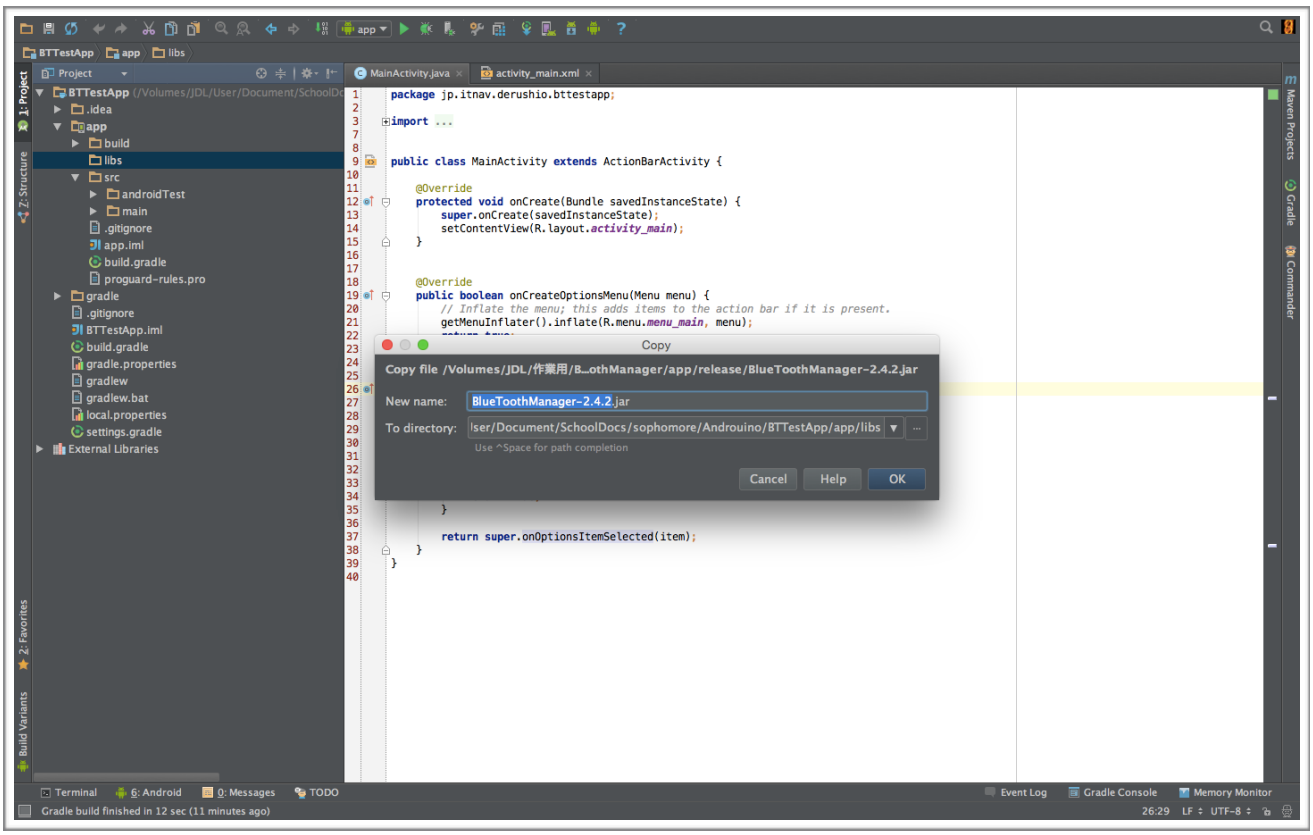

図１２ペーストしましょう

Android Studioの左下メニュー → ProjectからProject内のlibフォルダ（図１２参照）を選択し、”Ctrl + v”でペーストしましょう。

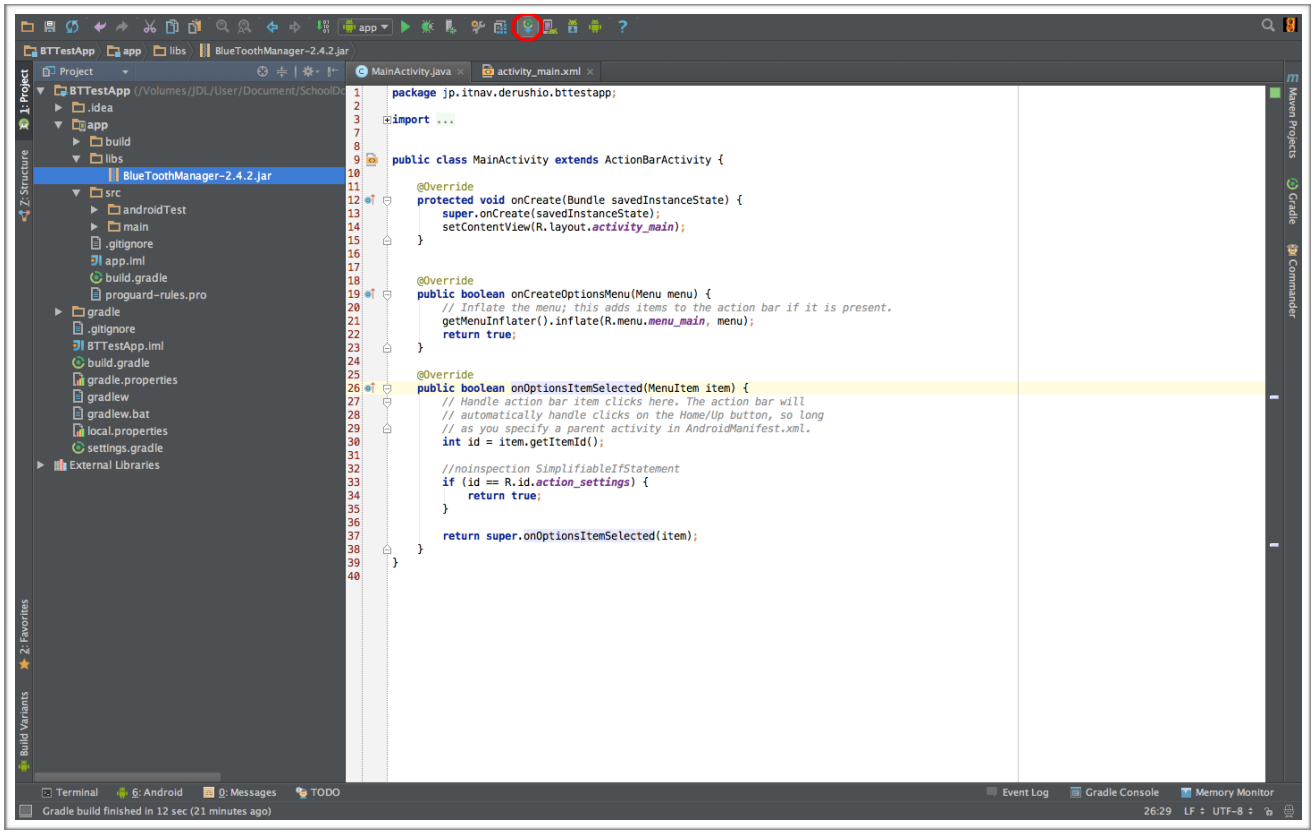

図１３リビルドしよう

図１３のようにクロームと↓が組み合わさったようなアイコンをクリックしてリビルドしましょう。

以上でライブラリを使う準備は整いました。しかし、実装方法については慣れていなければわからない部分が多いでしょう。そこで、サンプルプログラム(BluetoothManagerSample)をzip内に入れておいたので、それを参考にしながら実装していきましょう。

これからの説明で使用するソースファイルの場所

MainActivity.java

(展開したフォルダ)/BTManager-master/AndroidSample/BluetoothManagerSample/app/src/main/java/jp/itnav/derushio/bluetoothmanagersample/MainActivity.java

activity_main.xml

(展開したフォルダ)/BTManager-master/AndroidSample/BluetoothManagerSample/app/src/main/res/layout/activity_main.xml

AndroidManifest.xml

(展開したフォルダ)/BTManager-master/AndroidSample/BluetoothManagerSample/app/src/main/AndroidManifest.xml

MainActivity.java(直接ファイルを見ることを推奨)

```
public class MainActivity extends BluetoothManagedActivity {
        // BluetoothManagedActivityを継承している

        private LinearLayout paredDeviceList;
        // ペアリングしたBluetoothDeviceを表示するLinearLayout

        private TextView textViewReadMessage;
        // 受信したメッセージを表示するTextView
        private EditText editTextWriteMessage;
        // 送信するメッセージを設定するEditText

        private TimerHandler timerHandler;
        private boolean isTimerHandlerStarted = false;
        // タイマー

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);

                timerHandler = new TimerHandler();

                paredDeviceList = (LinearLayout) findViewById(R.id.paredDeviceList);
                textViewReadMessage = (TextView) findViewById(R.id.textViewReadMessage);
                editTextWriteMessage = (EditText) findViewById(R.id.editTextWriteMessage);
                // 描画された画面と変数を関連付け

                Set<BluetoothDevice> paredDevices = getParedDevices();
                // ペアリングされているデバイスを取得
                for (final BluetoothDevice bluetoothDevice : paredDevices) {
                        Button button = new Button(this);
                        // ボタンをインスタンス
                        button.setText(bluetoothDevice.getName() + "\n" + bluetoothDevice.getAddress());
                        button.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
                        button.setOnClickListener(new View.OnClickListener() {
                                @Override
                                public void onClick(View v) {
                                        setTargetDevice(bluetoothDevice);
                                        connectDevice();
                                        readMessageStart(100);
                                        timerHandler.timerStart(1000);
                                        isTimerHandlerStarted = true;
                                        // クリックされたら、クリックしたボタンに対応するBluetoothDeviceに接続、同時にメッセージの受信をスタートする。
                                }
                        });
                        paredDeviceList.addView(button);
                }
        }

        @Override
        protected void onPause() {
                super.onPause();
                if (isTimerHandlerStarted) {
                        timerHandler.timerStop();
                }
        }
        // 画面が止まった時にタイマーもストップ

        @Override
        protected void onResume() {
                super.onResume();
                if (isTimerHandlerStarted) {
                        timerHandler.timerStart();
                }
        }
        // 画面が始まった時にタイマーをスタート

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                // Inflate the menu; this adds items to the action bar if it is present.
                getMenuInflater().inflate(R.menu.menu_main, menu);
                return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
                // Handle action bar item clicks here. The action bar will
                // automatically handle clicks on the Home/Up button, so long
                // as you specify a parent activity in AndroidManifest.xml.
                int id = item.getItemId();

                //noinspection SimplifiableIfStatement
                if (id == R.id.action_settings) {
                        return true;
                }

                return super.onOptionsItemSelected(item);
        }

        public void writeMessage(View v) {
                editTextWriteMessage.clearFocus();
                writeMessage(editTextWriteMessage.getText().toString());
                // メッセージを送信する
        }

        private class TimerHandler extends Handler {
                // タイマーを定義するclass
                private boolean isTick = false;
                // タイマーが動いているか
                private long delayMilliSec = 1000;
                // 待ち時間の長さ

                @Override
                public void handleMessage(Message msg) {
                        super.handleMessage(msg);

                        if (isDeviceConnected()) {
                                textViewReadMessage.setText("返信：" + readMessage().get(0));
                                // メッセージを受信する。
                                // メッセージはArrayListの受信時間順に帰ってくるので、.get(0)で最新データが取れる。
                        }

                        if (isTick) {
                                sleep();
                        }
                }

                public void timerStart(long delayMilliSec) {
                        this.delayMilliSec = delayMilliSec;
                        isTick = true;
                        sleep();
                }

                public void timerStart() {
                        timerStart(delayMilliSec);
                }
                // リスタート用のOverLoad

                public void timerStop() {
                        isTick = false;
                }

                private void sleep() {
                        removeMessages(0);
                        sendMessageDelayed(obtainMessage(0), delayMilliSec);
                }
        }
}
```

上記コードで継承している自作クラス BluetoothManagedActivity要点

- protected void onCreate(android.os.Bundle savedInstanceState);
  - Bluetoothとタイマーを初期化します

- protected void onResume();
  - Bluetoothデバイスと接続しなおします

- protected void onPause();
  - Bluetoothデバイスと一旦切断します

- protected Set<BluetoothDevice> getParedDevices();
  - ペアリングされているBluetoothデバイス郡を取得します

- protected void setTargetDevice(BluetoothDevice targetDevice);
  - 接続するBluetoothデバイスをターゲットします

- protected BluetoothDevice getTargetDevice();
  - ターゲットされているBluetoothデバイスを取得します

- protected boolean isDeviceConnected();
  - Bluetoothデバイスに今つながっているか取得します

- protected void connectDevice();
  - Bluetoothデバイスに接続します

- protected void disConnectDevices();
  - Bluetoothデバイスから切断します

- protected void readMessageStart(long delayMilliSec);
  - delayMilliSecの秒数ごとにBluetoothからメッセージを受信し、メールボックスに蓄積します。

- protected void readMessageStop();
  - Bluetoothからメッセージを受信することを取りやめます

- protected void writeMessage(String message);
  - Bluetoothからメッセージを送信します

- protected ArrayList<String> readMessage();
  - readMessageStart(long delayMilliSec)で蓄積されているメールボックスを取得します。0が一番新しく、100件まで蓄積されます。

- protected void setOnConnect(Handler onConnect);
  - 接続時のハンドラを設定します。
  - message.whatについて
    - 0は成功
    - -1はSocketが見つからない
    - -2はBluetoothデバイスが見つからない
    - -3はそんなデバイスとペアリングしていない

- protected void setOnDisConnect(Handler onDisConnect);
  - 切断時のハンドラを設定します
  - message.whatについて
    - 0は成功
    - -1はBluetoothデバイスが見つからない

activity_main.xml(直接ファイルを見ることを推奨)

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context=".MainActivity">

        <ScrollView
                android:id="@+id/paredDeviceListScroll"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="1">

                <LinearLayout
                        android:id="@+id/paredDeviceList"
                        android:orientation="vertical"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"/>
        </ScrollView>

        <TextView
                android:id="@+id/textViewReadMessage"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="返信："
                android:textSize="24sp"
                android:textColor="@android:color/black"/>

        <EditText
                android:id="@+id/editTextWriteMessage"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"/>

        <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="送信"
                android:textSize="24sp"
                android:textColor="@android:color/black"
                android:onClick="writeMessage"/>
        <!-- クリックされたらwriteMessageメソッドを実行 -->


</LinearLayout>
```

一般的なレイアウトパーツを組み合わせてあるだけ。

AndroidManifest.xml(直接ファイルを見ることを推奨)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="jp.itnav.derushio.bluetoothmanagersample">

        <uses-permission android:name="android.permission.BLUETOOTH"/>
        <!-- Bluetooth機能を使用するので、許可設定をする -->

        <application
                android:allowBackup="true"
                android:icon="@drawable/ic_launcher"
                android:label="@string/app_name"
                android:theme="@style/AppTheme">
                <activity
                        android:name=".MainActivity"
                        android:label="@string/app_name">
                        <intent-filter>
                                <action android:name="android.intent.action.MAIN"/>

                                <category android:name="android.intent.category.LAUNCHER"/>
                        </intent-filter>
                </activity>
        </application>

</manifest>
```

Bluetooth機能をつかうので、パーミッション設定でBLUETOOTHを宣言する

最後に

ここまであらかたライブラリについての説明を書いてきましたが、真の意味でこのライブラリを使いこなしたければ、ソースコードも含まれていますのでそちらも参照してください。