

远望谷XC1003手机读写文档说明

该文档是针对客户使用的文档，如果有什么疑惑、问题，请联系相关技术人员。

读取标签TID

读取标签的Tid区域，直接调用`public void readTid(ReadTagDataListenser listenser)`或者`public void readTid(int length, ReadTagDataListenser listenser)`其中`ReadTagDataListenser`表示结果的回调，`length`表示读取Tid区域的长度。

备注：标签的Tid区域有96位和128为之分。在程序中体现为长度是24和32个十六进制数。而在程序的底层，又将每4个十六进制数分为1组。所以如果想要读取一个Tid为128位标签的所有Tid数据，那么需要调用`readTid(8, listenser)`来读取所有的Tid数据。

以下是读取一枚Tid长度为96标签的Tid例子：

```
readTid(new ReadTagLastActivity.ReadTagDataListenser() {  
  
    @Override  
  
    public void onSuccess(String data) {  
  
        //读取标签成功，请将标签数据  
  
        tv_reasult.setText(data);  
  
    }  
})
```

```

@Override

public void onFailed(String msg, int errorCode) {

    tv_reasult.setText(msg+"***code***"+errorCode);

}

});

```

读取标签Epc

读取标签的eps区域使用readEpc(String pwd, ReadTagDataListenser listenser) 这个方法即可，第一个参数代表访问密码，第二个参数为读取标签的回调。

一个标签如果设置了访问密码，那么在调用该方法的时候，如果传入了错误密码是不可以读取到数据的。但是！如果传入正确的密码或者传入**空的密码**都可以读取到标签的。就是给第一个参数传入一个为空的字符串是什么时候都可以读取到标签的。

以下是读取epc的例子

```

readEpc(et_count.getText().toString().trim(), new ReadTagDataListenser() {

@Override

public void onSuccess(String data) {

    tv_reasult.setText(data);

}

});

```

```

    }

    @Override

    public void onFailed(String msg, int errorCode) {

        tv_reasult.setText(msg+"*****code***"+errorCode);

    }

});

```

读取User区域

读取标签的user区域，调用readUser(int readLength, ReadTagDataListenser listenser)
该方法的第一个参数和readTid中的length使用方法一样，第二个参数为读取标签的回调。

读取user区域不进行赘述，需要强调一点，在某些便签中，是没有user区域的，如果发生读取不到的情况，请确保该标签含有user区域。

一下为读取user区域的例子：

```

readUser(count, new ReadTagLastActivity.ReadTagDataListenser() {

    @Override

    public void onSuccess(String data) {

        tv_reasult.setText(data);
    }
});

```

```

    }

    @Override

    public void onFailed(String msg, int errorCode) {

        tv_result.setText(msg);

    }

});

```

写入数据到Epc

写入数据到epc，调用writeEpc(int offset, String data, String pwd, WriteTagDataListenser listenser)，第一个参数为跳过的数据，第二个参数为写入的数据，第三个参数为写入的密码，最后一个参数为写入标签的回调。

以下为使用的案例：

```

//调用写入的方法

writeEpc(be, data, pwd, new WriteTagDataListenser() {

    @Override

    public void onSuccess() {

        Toast.makeText(WriteEpcDataActivity.this, "写入成功",
        Toast.LENGTH_SHORT).show();
    }
});

```

```

        handler.sendMessageDelayed(1,500);

    }

    @Override

    public void onFailed(String msg, int errorCode) {

        tv_result.setText(msg+"读取结果: " + errorCode);

    }

});

```

写入数据到user

将数据写入到epc，调用 `writeUser(int offset, String data, String pwd, WriteTagDataListenser listener)`，第一个参数为跳过的数据，第二个参数为写入的数据，第三个参数为写入的密码，最后一个参数为写入标签的回调。

注意核实该标签是否含有User区域。

以下为使用案例

```

writeUser(be, data, pwd, new WriteTagDataListenser() {

    @Override

    public void onSuccess() {

```

```
        Toast.makeText(WriteUserDataActivity.this,"写入成功",Toast.LENGTH_SHORT).show();
```

```
        handler.sendMessageDelayed(1,500);  
  
    }
```

```
@Override
```

```
public void onFailed(String msg, int errorCode) {
```

```
}
```

```
});
```