



(C) SCL 为低电平时拉高 SDA

9、以下为 STM32 的 GPIO 端口配置寄存器的描述，在 GPIO 控制 LED 电路设计时，要使最大输出速度为 10MHz，应该设置 ( )

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	<b>CNFy[1:0]:</b> 端口x配置位(y = 8...15) (Port x configuration bits) 软件通过这些位配置相应的I/O端口，请参考表17端口位配置表。 在输入模式(MODE[1:0]=00): 00: 模拟输入模式 01: 浮空输入模式(复位后的状态) 10: 上拉/下拉输入模式 11: 保留 在输出模式(MODE[1:0]>00): 00: 通用推挽输出模式 01: 通用开漏输出模式 10: 复用功能推挽输出模式 11: 复用功能开漏输出模式
位9:28 25:24 21:20 17:16 13:12 9:8, 5:4 1:0	<b>MODEy[1:0]:</b> 端口x的模式位(y = 8...15) (Port x mode bits) 软件通过这些位配置相应的I/O端口，请参考表17端口位配置表。 00: 输入模式(复位后的状态) 01: 输出模式，最大速度10MHz 10: 输出模式，最大速度2MHz 11: 输出模式，最大速度50MHz

- (A) CNFy[1:0] (B) MODEy[1:0]  
(C) MODE (D) CNF

10、PWM 是( )

- (A) 脉冲宽度调制 (B) 脉冲频率调制 (C) 脉冲幅度调制 (D) 脉冲位置调制

11、Linux 中对目标文件赋予所有者和同组可读可写不可执行、其他用户可读不可写不可执行的权限的命令为:

- (A) chmod 755 (B) chmod 644 (C) chmod 655 (D) chmod 744

12、下列选项中不需在 AndroidManifest.xml 中设置或注册的是 ( )。

- (A) 数据库扩展类的信息 (B) Service 的名称  
(C) Activity 的名称和信息 (D) 所需权限的声明

13、利用 AlertDialog 建立列表对话框时，需要调用 setItems 方法，并传入包含条目文本的字符串数组以及( )

- (A) 确定、取消按钮的点击响应 (B) 对话框标题  
(C) 包含默认选择情况的布尔型数组 (D) 条目的点击响应

14、以下哪一个不属于 Android 项目的组件

- (A) Intent (B) ContentProvider (C) Service (D) Activity

15、Activity 的生命周期函数不包括哪一个

- (A) onStart (B) onBackPressed (C) onResume (D) onSop

## 二 填空题 (每空 2 分，共 20 分)

1、STM32 的所有端口都有外部中断能力。当使用\_\_\_\_\_时，相应的引脚必须配置

成\_\_\_\_\_。

2、STM32 具有单独的位设置或位清除能力。这是通过\_\_\_\_\_和\_\_\_\_\_寄存器来实现的。

3、系统计时器（SysTick）提供了 1 个\_\_\_\_\_位精度的计数器，具有灵活的控制机制

4、在 STM32 中，从外设(TIMx、ADC、SPIx、I2Cx 和 USARTx)产生的 7 个请求，通过逻辑\_\_\_\_\_输入到 DMA 控制器，这样同时\_\_\_\_\_个请求有效。

5、STM32 中常用到的三种串行通信协议分别是\_\_\_\_\_、\_\_\_\_\_、和\_\_\_\_\_。

6、使用 startService 启动的 intentService 其目标执行内容应写在其内部的\_\_\_\_\_方法内。

7、安卓提供了访问 Sqlite 数据库的接口类是\_\_\_\_\_，使用时通过复写其中的 onCreate 函数完成表格结构的定义。

8、当使用 Intent 的启动 SecondActivity 时，如需返回结果，应使用函数\_\_\_\_\_，并传入此 Intent 以及相应的 requestCode。

9、当对文件进行写操作时，可以对目标文件的 File 类使用\_\_\_\_\_方法建立流。

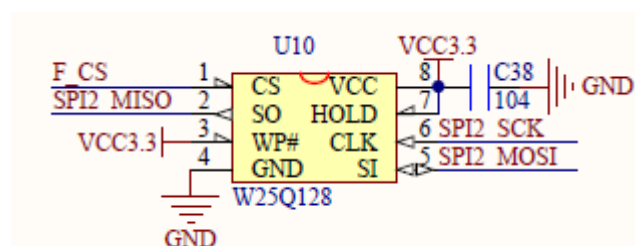
10、要让目标文本显示框显示内容，应对其使用\_\_\_\_\_方法，传入欲显示内容的字符串对象。

### 三 编程题（共 50 分）

1、（20 分）以下图分别为 W25Q128 部分指令集，部分电路原理图。请编写程序，使用 SPI 协议读写 W25Q128。以下为部分程序，请补齐缺失的部分。

Data Input Output	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Clock Number	(0 - 7)	(8 - 15)	(16 - 23)	(24 - 31)	(32 - 39)	(40 - 47)	(48 - 55)
Release Power-down / ID	ABh	Dummy	Dummy	Dummy	(ID7-ID0) <sup>(2)</sup>		
Manufacturer/Device ID	90h	Dummy	Dummy	00h	(MF7-MF0)	(ID7-ID0)	
JEDEC ID	9Fh	(MF7-MF0)	(ID15-ID8)	(ID7-ID0)			

F CS	PB12	73	PB11/I2C2_SDA/USART3_RX
SPI2 SCK	PB13	74	PB12/SPI2_NSS/I2S2_WS/I2C2_SMBAL/US
SPI2 MISO	PB14	75	PB13/SPI2_SCK/I2S2_CK/USART3_CTS/TI
SPI2 MOSI	PB15	76	PB14/SPI2_MISO/USART3_RTS/TIM1_CH
			PB15/SPI2_MOSI/I2S2_SD/TIM1_CH3N



```
void W25QXX_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd( (1_____), ENABLE );

    GPIO_InitStructure.GPIO_Pin = (2_____);
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
```

```

GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);
GPIO_SetBits(GPIOB, GPIO_Pin_12);

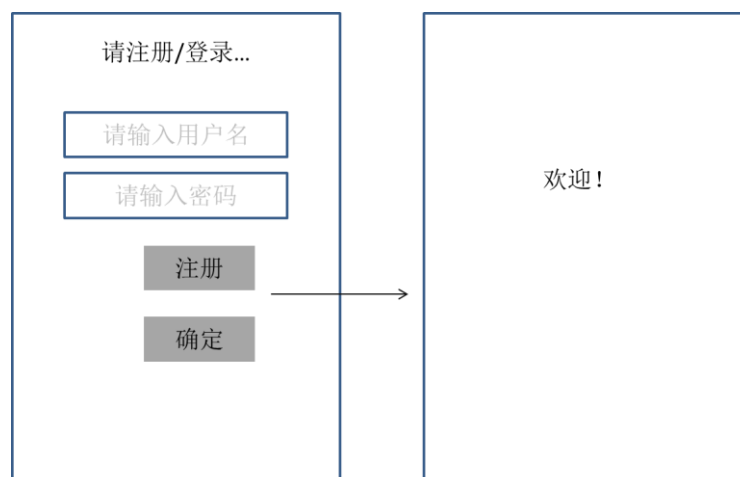
W25QXX_CS=1;
SPI2_Init();
SPI2_SetSpeed(SPI_BaudRatePrescaler_2)
W25QXX_TYPE=W25QXX_ReadID();

}

u16 W25QXX_ReadID(void)
{
    u16 Temp = 0;
    W25QXX_CS=(3_____);
    SPI2_ReadWriteByte( 4_____);//发送读取 ID 命令
    SPI2_ReadWriteByte(0x00);
    SPI2_ReadWriteByte(0x00);
    SPI2_ReadWriteByte(0x00);
    Temp|=(5_____);
    Temp|=SPI2_ReadWriteByte(0xFF);
    W25QXX_CS=1;
    return Temp;
}

```

2、（30 分）简易登录界面。假设登录用户名、密码等信息已存入数据库 myData,表名 myTable,其列名为主键 \_ID, 用户名 NAME 以及密码 PASSWORD。 请完成如图界面，并填充代码，实现简易注册、登录功能。注意：对于标号相同的空格，任填其中一个即可。



布局 1 代码:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:_____6_____>    <!--设置为从上到下线性布局-->
    <TextView android:text = “请注册/登录...”/>

```

```

<EditText android:hint = “请输入用户名”
            android:id = “_____7_____ /et1”/>  <!--添加 ID-->
<EditText android:hint = “请输入密码”
            android:id = “_____7_____ /et2”/>
<Button android:text = “注册”
            android:id = “_____7_____ /bt1”/>
<Button android:text = “登录”
            android:id = “_____7_____ /bt2”/>
<LinearLayout/>

```

*myData 代码:* (自定义数据库, 用于存放密码、用户名, 略)

*布局2 代码:* (略)

*Activity2 代码:* (登陆后界面, 略)

*Activity1 代码:*

imports (略)

```

Public class Activity1 extends Activity{
private EditText et1,et2;
private Button bt1,bt2;
private boolean isNameExist=false, isPasswordCorrect=false;

```

@Override

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    initView();//通过查找 id 方式获得控件
    setEvent();//绑定点击事件
}

```

```

initView(){
    _____8_____ //查找布局中已定义的控件 (任写一个)
}

```

```

setEvent(){
    myData helper = new myData(Activity1.this);
    SQLiteDatabase database = _____9_____ //通过 helper 获取数据库访问接口
    String str1 = _____10_____
    String str2 = _____10_____ //二者任填其一

```

```

    Cursor cursor = _____11_____ //查找数据库中所有元素,由 cursor 接收
    for(int i=0; i<cursor.getCount();i++){
        if(_____12_____.equals(str1)){//若数据库中已存在输入的用户名
            isNameExist = true;
            if(_____12_____.equals(str2)){//若对应密码与数据库中一致
                isPasswordCorrect = true;
            }
        }
    }
}

```

```

        }
    }
}
bt1.setOnClickListener(new view.OnClickListener{///按钮 1 点击事件
    public void onClick(){
        if(isNameExist==true){
            Toast.makeText(Activity1.this,"用户已存在",1).show();
        }else{
            ContentValues value1 = new ContentValues();//声明并实例化
            value1.put("NAME",str1);value1.put("PASSWORD", str2);
            _____13_____//将新用户信息 insert 入数据库
        }
    }
}
}
bt2.setOnClickListener(new view.OnClickListener{///按钮 2 点击事件
    public void onClick(){
        if(isNameExist==true && isPasswordCorrect==true){
            _____14_____//使用 Intent 跳转到 Activity2
        }
    }
}
_____15_____//关闭对数据库访问
}
}

```