

08 嵌入式系统开发 设计案例



目录

CONTENTS

8.1 嵌入式综合实例分析

8.2 系统设计

8.3 QT界面设计



8.3 QT界面设计--环境监测系统实例分析

- 设计背景

随着实际生活中，生活水平的提高，人们对环境越来越敏感。越来越多的智能家居、智能农业系统出现，改变了生活方式。而在环境因素中，最多的就是温度、湿度、光照、PM2.5等。本项目设计一个室内温湿度监测系统。

8.3 QT界面设计--环境监测系统实例分析

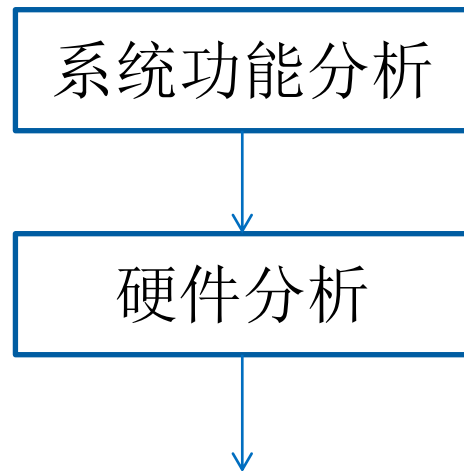
- 设计背景





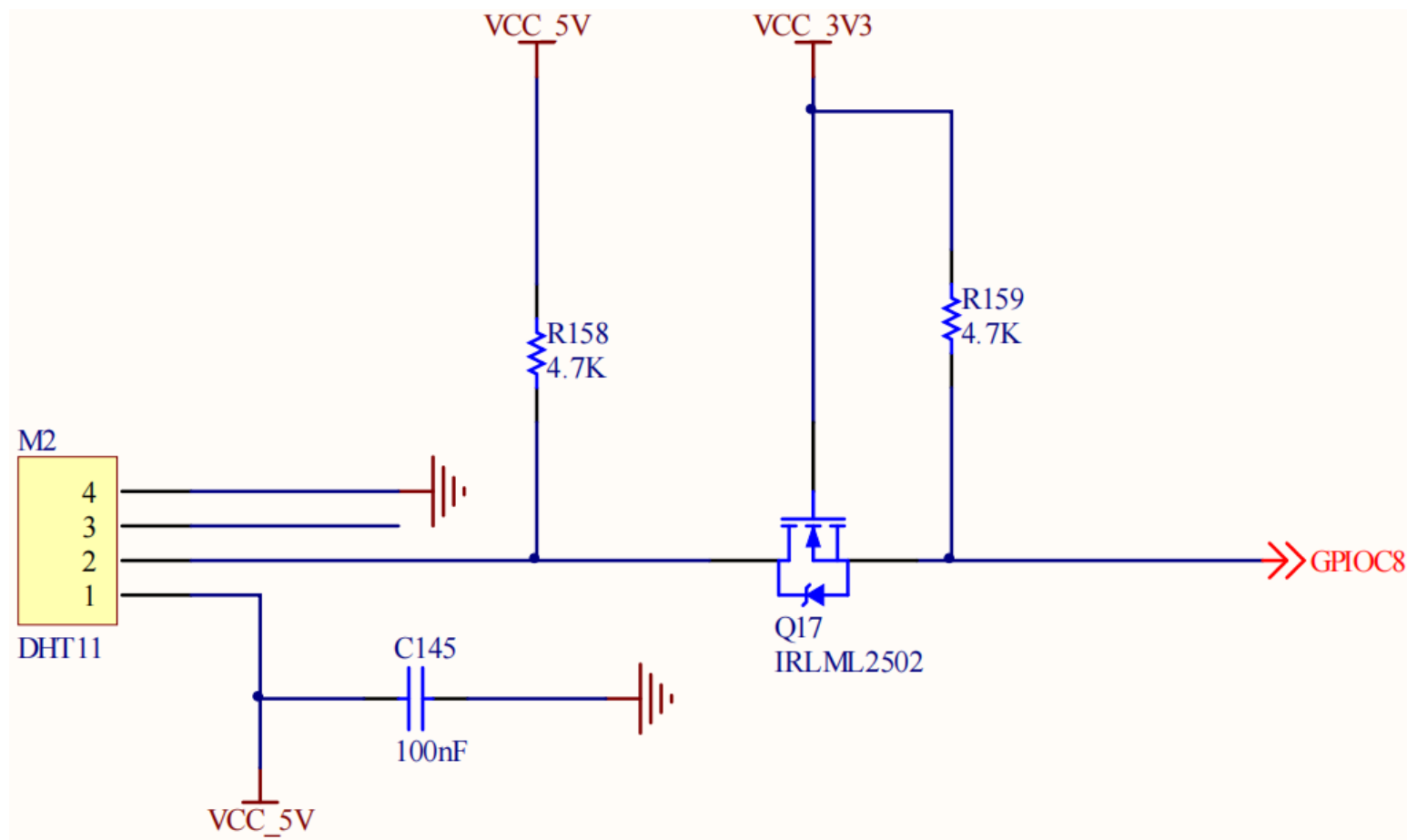
8.3 QT界面设计--环境监测系统实例分析

- **项目流程**





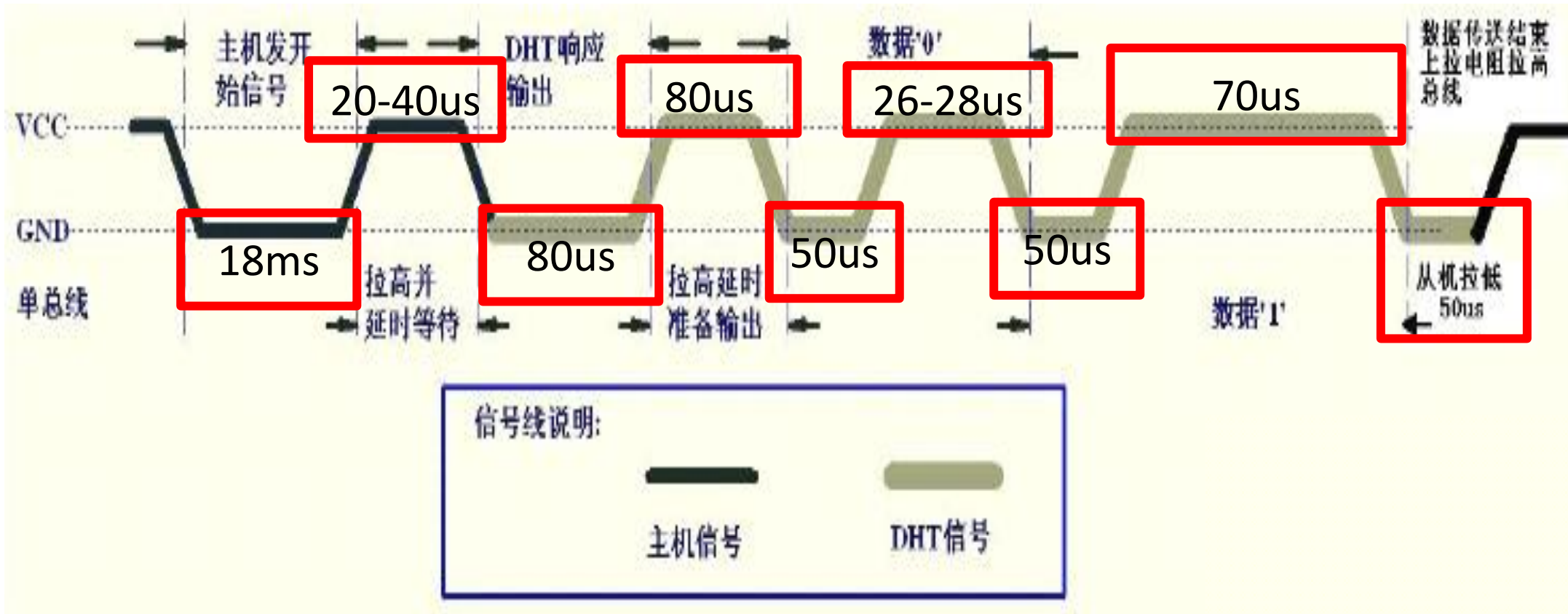
8.3 QT界面设计--环境监测系统实例分析



DHT11温度传感器电路图



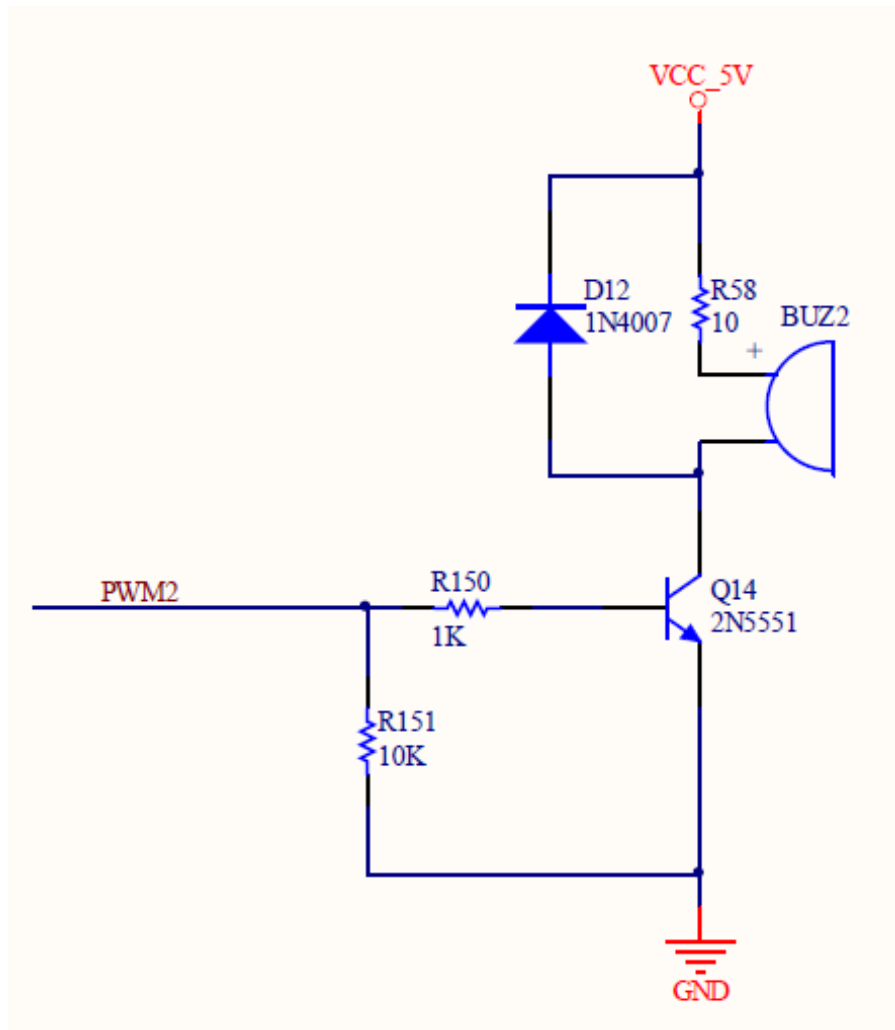
8.3 QT界面设计--环境监测系统实例分析



DHT11时序图



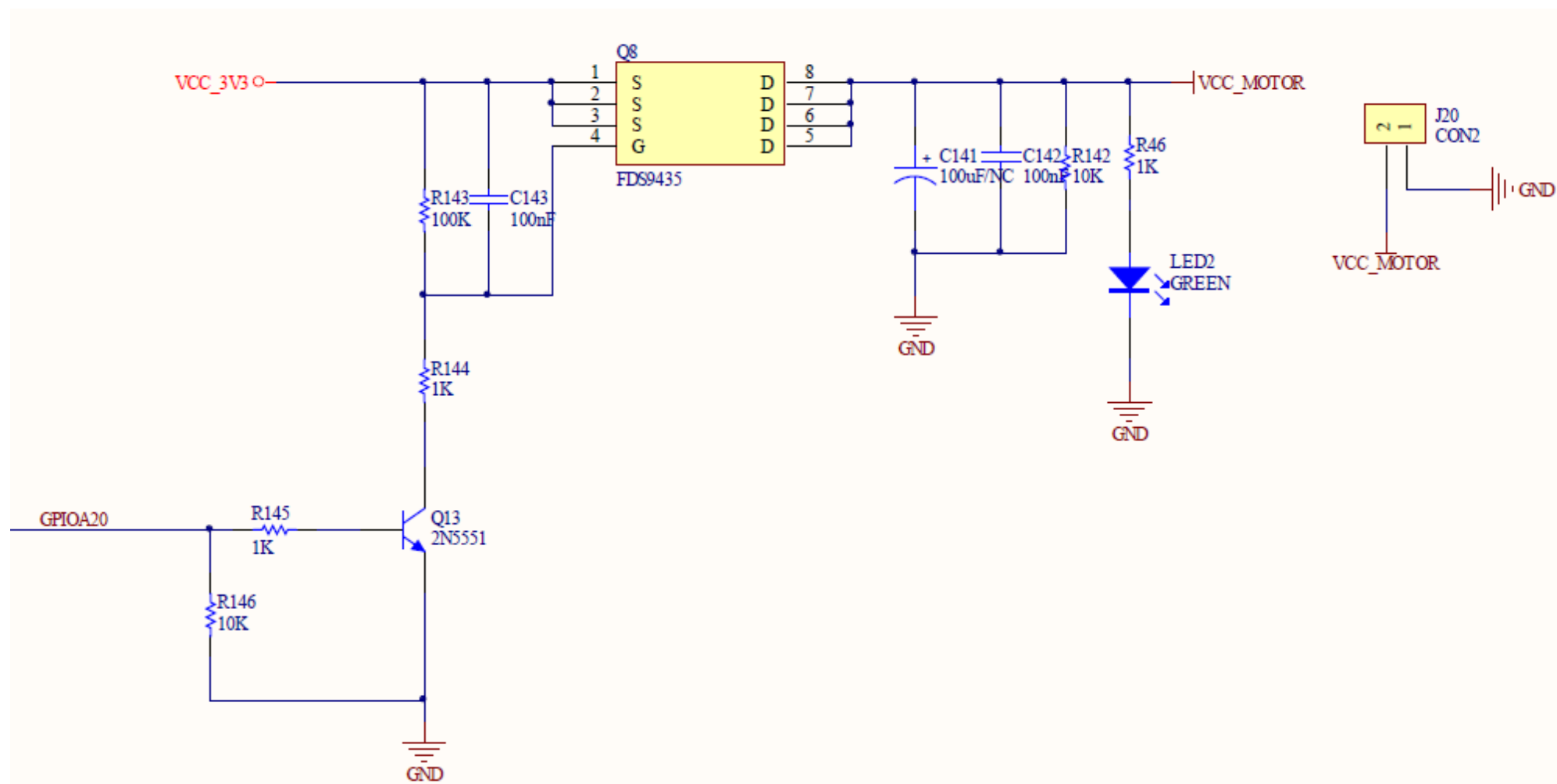
8.3 QT界面设计--环境监测系统实例分析



蜂鸣器电路图



8.3 QT界面设计--环境监测系统实例分析

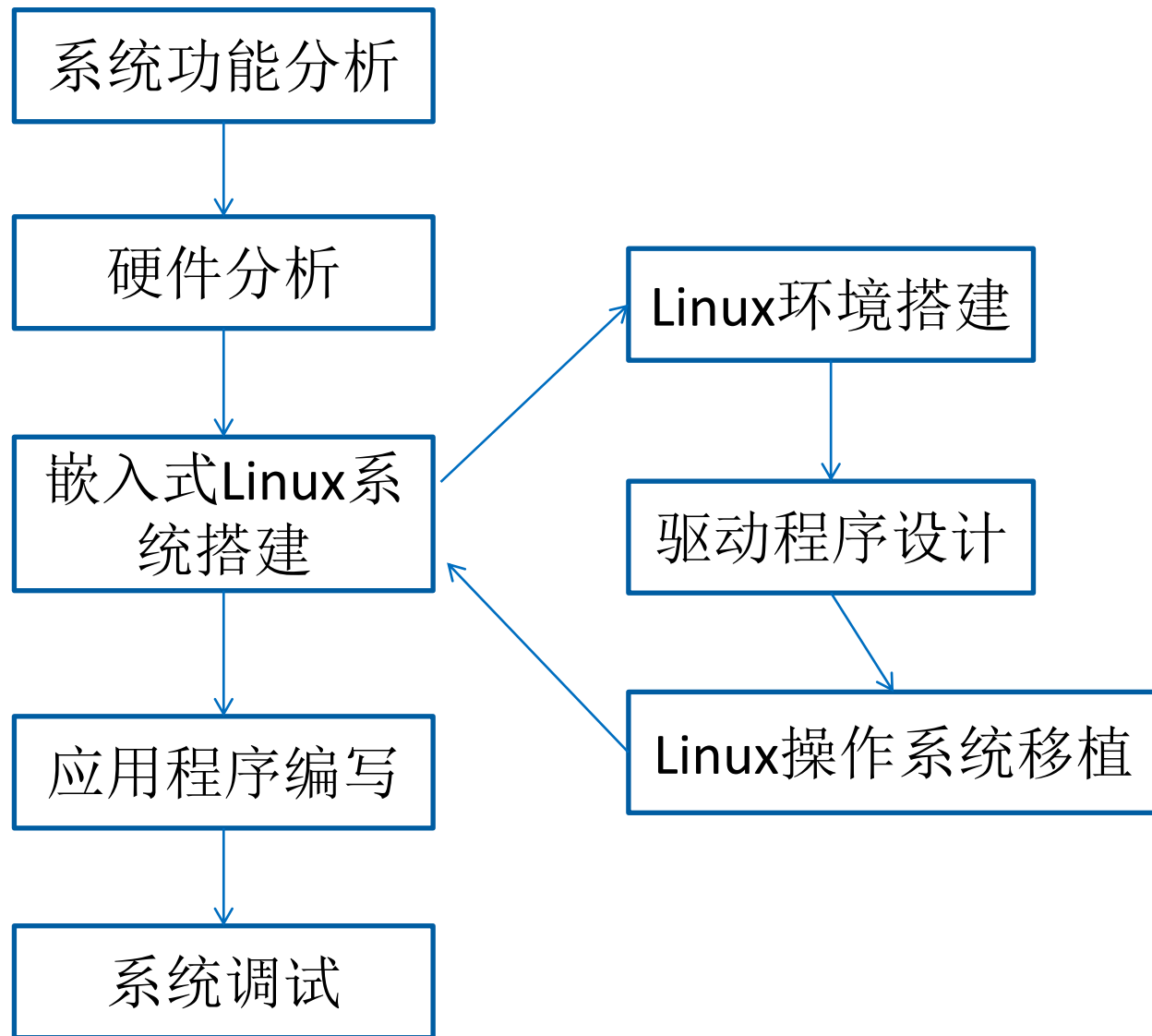


直流电机电路图



8.3 QT界面设计--环境监测系统实例分析

- 项目流程





8.3 QT界面设计—驱动程序设计

DHT11驱动设计：

```
static struct file_operations cvt4418_dht11_fops = {  
    .owner = THIS_MODULE,  
    .release = dht11_sensor_release,  
    .open = dht11_sensor_open,  
    .read = dht11_sensor_read,  
};
```



8.3 QT界面设计—驱动程序设计

```
.release = dht11_sensor_release,
```

```
static int dht11_sensor_release(struct inode *inode, struct file *filp)
```

```
{
```

```
    module_put(THIS_MODULE);
```

```
    dht11_dev.lock = 0;
```

```
    return 0;
```

```
}
```



8.3 QT界面设计—驱动程序设计

```
.open = dht11_sensor_open,
```

```
static int dht11_sensor_open(struct inode *inode, struct file *filp)
```

```
{
```

```
    if(dht11_dev.lock)
```

```
        return -EBUSY;
```

```
    try_module_get(THIS_MODULE);
```

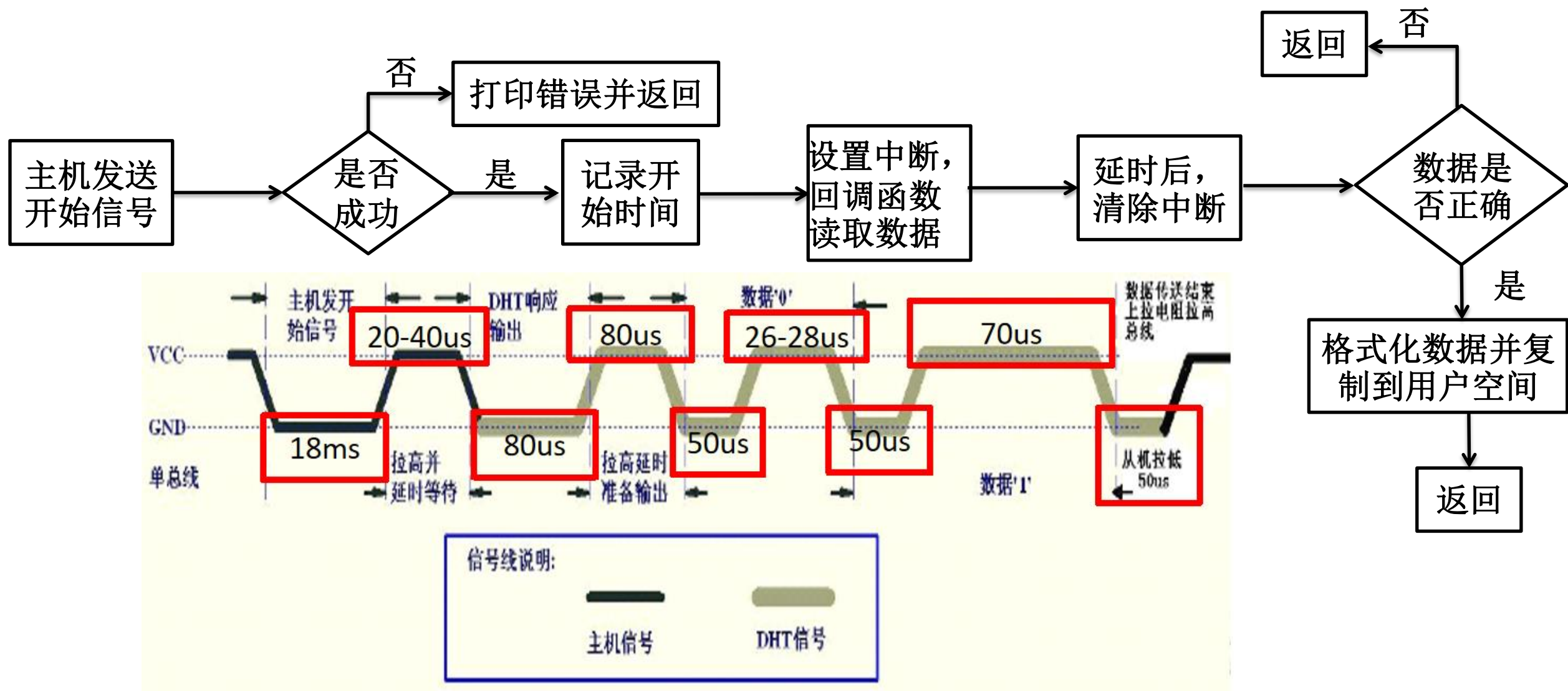
```
    dht11_dev.lock = 1;
```

```
    return 0;
```

```
}
```

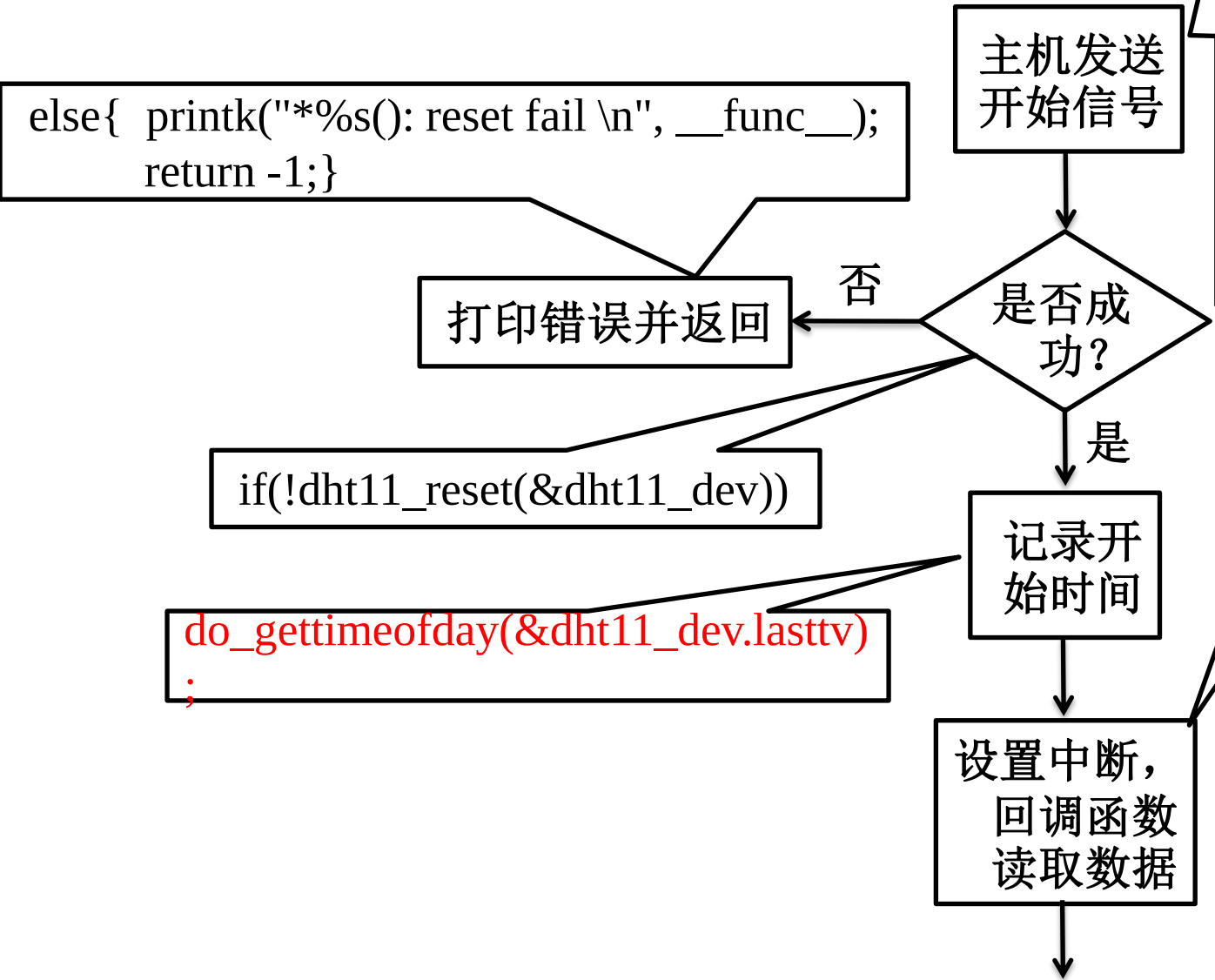
8.3 QT界面设计—驱动程序设计

```
.read = dht11_sensor_read(struct file *filp,char __user *buffer,size_t count,loff_t *f_pos)
```





8.3 QT界面设计—驱动程序设计



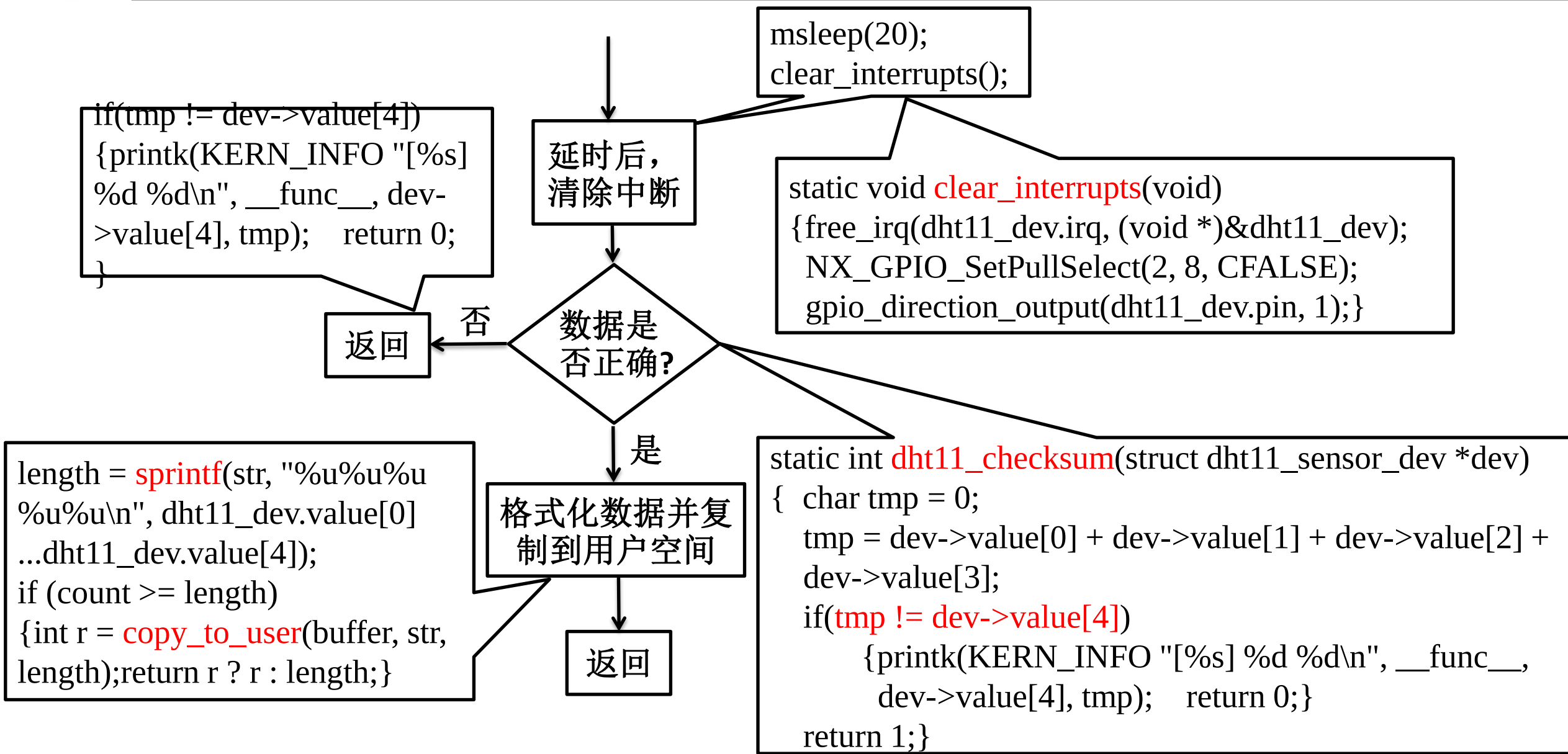
```
static int dht11_reset(struct dht11_sensor_dev *dev)
{
    gpio_direction_output(dev->pin, 1);
    dev->write_bit(dev->pin, 0);
    msleep(18);
    dev->write_bit(dev->pin, 1);
    gpio_direction_input(dev->pin);
    dev->delay(20);
    return 0;
}
```

```
setup_interrupts();

static int setup_interrupts(void)
{
    int result; dht11_dev.irq =
    gpio_to_irq(dht11_dev.pin);
    result = request_irq(dht11_dev.irq,
    dht11_interrupt, IRQ_TYPE_EDGE_BOTH, DEVICE_NAME, (void *)&dht11_dev);
    switch (result)
    {
        .....
    }
    return 0;
}
```



8.3 QT界面设计—驱动程序设计





8.3 QT界面设计—QT概述

QT是一个跨平台的C++图形用户界面库，基于面向对象的设计思想，提供了信号（signal）与槽（slot）的通信机制，具有可查询，可设计的属性，以及强大的事件和事件过滤器。





8.3 QT界面设计—QT概述

产生历史：

- 1991年 QT最早由奇趣科技公司开发。
- 1996年 进入商业领域，是目前linux桌面KDE的基础。
- 2008年 奇趣被诺基亚公司收购，成为诺基亚旗下的编程工具。
- 2012年 QT被IT公司Digia收购。
- 2014年 跨平台集成开发环境QT Creator 3.1.0发布，从此逐步实现了对多平台支持。



8.3 QT界面设计—QT概述

支持平台：

- ✓ windows xp、 7、 8、 10
- ✓ Linux、 FreeBSD
- ✓ Mac os
- ✓ Win CE



8.3 QT界面设计—QT概述

成功案例：

- Linux桌面环境KDE
- WPS Office
- Skype网络电话
- Google Earth
- VLC播放器
-





8.3 QT界面设计—QT概述

QT安装：<https://download.qt.io/archive/qt/>

Name		Last modified	Size	Metadata
↑ Parent Directory			-	
📁 5.14/		31-Mar-2020 09:14	-	
📁 5.13/		31-Oct-2019 07:17	-	
📁 5.12/		07-Apr-2020 07:36	-	
📁 5.9/	Name	Last modified	Size	Metadata
📁 5.1/	↑ Parent Directory		-	
📁 5.0/	📁 submodules/	31-Mar-2020 09:27	-	
📁 4.8/	📁 single/	31-Mar-2020 10:10	-	
📁 4.7/	📄 qt-opensource-windows-x86-5.14.2.exe	31-Mar-2020 10:18	2.3G	Details
	📄 qt-opensource-mac-x64-5.14.2.dmg	31-Mar-2020 10:16	2.6G	Details
	📄 qt-opensource-linux-x64-5.14.2.run	31-Mar-2020 10:14	1.2G	Details

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/218000073050006040>