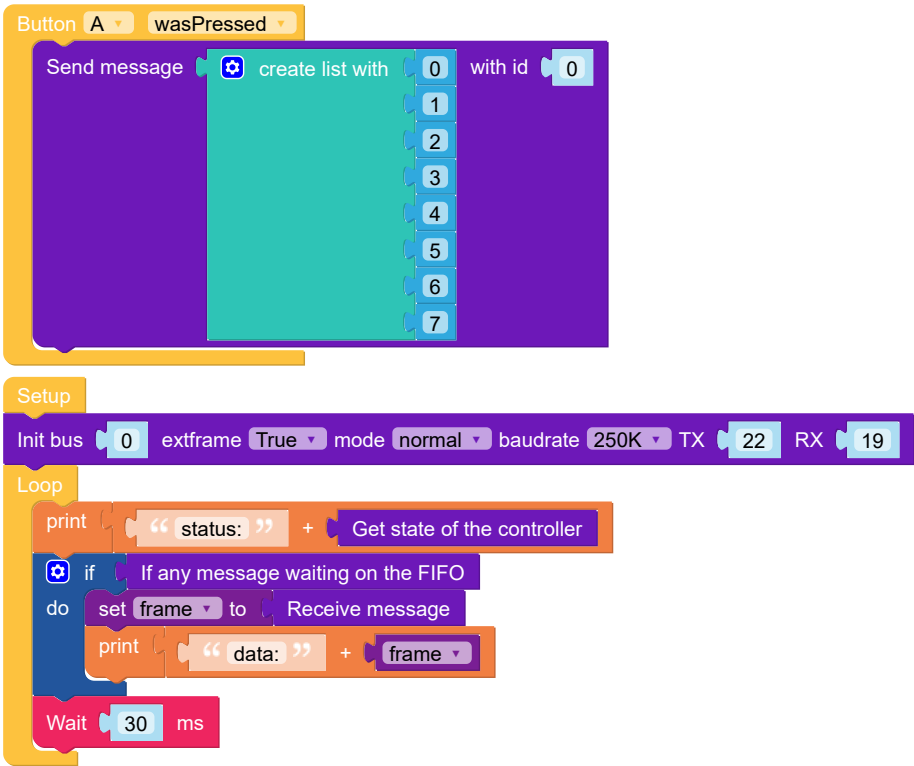


Atomic CAN Base

| 案例程序

按钮A按下时发送CAN数据帧 [0, 1, 2, 3, 4, 5, 6, 7] (ID为0) , 并在主循环中持续检测CAN总线状态



```

from m5stack import *
from m5ui import *
from uiflow import *
from base.CAN import CAN
import time
frame = None
atom_can = CAN()

def buttonA_wasPressed():
    global frame
    atom_can.send([0, 1, 2, 3, 4, 5, 6, 7], 0)
    pass
btnA.wasPressed(buttonA_wasPressed)

atom_can.can_init(0, extframe=True, mode=atom_can.NORMAL, baudrate=atom_can.BAUDRATE_250K,
tx_io=22, rx_io=19, auto_restart=False)
while True:
    print((str('status:') + str(atom_can.state()))))
    if atom_can.any():
        frame = atom_can.recv()
        print((str('data:') + str(frame)))
    wait_ms(30)
    wait_ms(2)

```

Base

功能说明

Init bus extframe mode baudrate TX RX

```

from base.CAN import CAN
atom_can = CAN()
atom_can.can_init(0, extframe=True, mode=atom_can.NORMAL, baudrate=atom_can.BAUDRATE_250K,
tx_io=22, rx_io=19, auto_restart=False)

```

- 初始化 CAN 总线，配置是否为拓展帧模式，工作模式(常规模式，回环模式等)以及波特率设置

If any message waiting on the FIFO

```
atom_can.any()
```

- 检查 FIFO 中是否有未读的数据

Receive message

```
frame = atom_can.recv()
```

- 接收数据

Send message with id 0

```
atom_can.send([0, 1, 2, 3, 4, 5, 6, 7], id)
```

- 发送一条数据，并指定数据帧的 ID, ID 长度为1个 byte, 传入的数据类型要求为list或是tuple, 数据帧的数据长度要求是8个 byte。

Set filter bank 0 mode filter_raw_single message

```
atom_can.setfilter(0, CAN.FILTER_RAW_SINGLE, [])
```

- 设置过滤组

Get state of the controller

```
atom_can.state()
```

- 获取 CAN 控制器状态

Clear RX queue

```
atom_can.clear_rx_queue()
```

- 清除接收队列

Clear TX queue

```
atom_can.clear_tx_queue()
```

- 清除发送队列

Clear filter

```
atom_can.clearfilter()
```

- 清除过滤组

Restart

```
atom_can.restart()
```

- 重启 CAN 总线

Deinit

```
atom_can.deinit()
```

- 停止 CAN 总线