

```

#ifndef _CAN2040_H
#define _CAN2040_H

//
// The C++world needs this ...
//
#ifdef __cplusplus
extern "C" {
#endif

#include <stdint.h> // uint32_t

//
// Option to enable the experimental multi core support.
//
#define CAN_PICO_MULTI_CORE 1

#if CAN_PICO_MULTI_CORE == 1
#include "pico/lock_core.h"
#endif

struct can2040_msg {
    uint32_t id;
    uint32_t dlc;
    union {
        uint8_t data[8];
        uint32_t data32[2];
    };
};

enum {
    CAN2040_ID_RTR = 1<<30,
    CAN2040_ID_EFF = 1<<31,
};

enum {
    CAN2040_NOTIFY_RX = 1<<20,
    CAN2040_NOTIFY_TX = 1<<21,
    CAN2040_NOTIFY_ERROR = 1<<23,
};

struct can2040;
typedef void (*can2040_rx_cb)(struct can2040 *cd, uint32_t notify
    , struct can2040_msg *msg);

void can2040_setup(struct can2040 *cd, uint32_t pio_num);
void can2040_callback_config(struct can2040 *cd, can2040_rx_cb rx_cb);
void can2040_start(struct can2040 *cd, uint32_t sys_clock, uint32_t bitrate
    , uint32_t gpio_rx, uint32_t gpio_tx);
void can2040_shutdown(struct can2040 *cd);

```

```
void can2040_pio_irq_handler(struct can2040 *cd);
int can2040_check_transmit(struct can2040 *cd);
int can2040_transmit(struct can2040 *cd, struct can2040_msg *msg);
```

```
/******
```

```
* Internal definitions
```

```
******/
```

```
struct can2040_bitunstuffer {
    uint32_t stuffed_bits, count_stuff;
    uint32_t unstuffed_bits, count_unstuff;
};
```

```
struct can2040_transmit {
    struct can2040_msg msg;
    uint32_t crc, stuffed_words, stuffed_data[5];
};
```

```
// a central structure to keep all configuration data...
```

```
struct can2040ConfigDesc {

    uint32_t      mcPioNum;
    uint32_t      mcSysClock;
    uint32_t      mcBitRate;
    uint32_t      mcRxPin;
    uint32_t      mcTxPin;
    can2040_rx_cb mcRxCallback;
    uint32_t      mcRxQueueSize;
    bool          mcRunOnCore1;
    bool          mcSetupOK;
};
```

```
struct can2040 {
```

```
#if CAN_PICO_MULTI_CORE == 1
```

```
    // need a lock for protecting the CAN transmit queue
```

```
    lock_core_t    tx_queue_lock;
```

```
#endif
```

```
    // Setup
```

```
    uint32_t pio_num;
```

```
    void *pio_hw;
```

```
    uint32_t gpio_rx, gpio_tx;
```

```
    can2040_rx_cb rx_cb;
```

```
    // Bit unstuffing
```

```
    struct can2040_bitunstuffer unstuf;
```

```
    uint32_t raw_bit_count;
```

```
// Input data state
uint32_t parse_state;
uint32_t parse_crc, parse_crc_bits, parse_crc_pos;
struct can2040_msg parse_msg;

// Reporting
uint32_t report_state;

// Transmits
uint32_t tx_state;
uint32_t tx_pull_pos, tx_push_pos;
struct can2040_transmit tx_queue[4];
};

//
// The C++world needs this ...
//
#ifdef __cplusplus
}
#endif

#endif // can2040.h
```