/\*\*

\* Library extends MFRC522.h to support RATS for ISO-14443-4 PICC.

\* RATS - Request for Answer To Select.

\* @author JPG-Consulting

\*/

#ifndef MFRC522Extended\_h

#define MFRC522Extended\_h

#include <Arduino.h>

#include "MFRC522.h"

class MFRC522Extended : public MFRC522 {

public:

// ISO/IEC 14443-4 bit rates

enum TagBitRates : byte {

BITRATE\_106KBITS = 0x00,

BITRATE\_212KBITS = 0x01,

BITRATE\_424KBITS = 0x02,

BITRATE\_848KBITS = 0x03

};

// Structure to store ISO/IEC 14443-4 ATS

typedef struct {

byte size;

byte fsc; // Frame size for proximity card

struct {

bool transmitted;

bool sameD; // Only the same D for both directions supported

TagBitRates ds; // Send D

TagBitRates dr; // Receive D

} ta1;

struct {

bool transmitted;

byte fwi; // Frame waiting time integer

byte sfgi; // Start-up frame guard time integer

} tb1;

struct {

bool transmitted;

bool supportsCID;

bool supportsNAD;

} tc1;

// Raw data from ATS

byte data[FIFO\_SIZE - 2]; // ATS cannot be bigger than FSD - 2 bytes (CRC), according to ISO 14443-4 5.2.2

} Ats;

// A struct used for passing the PICC information

typedef struct {

uint16\_t atqa;

Uid uid;

Ats ats;

// For Block PCB

bool blockNumber;

} TagInfo;

// A struct used for passing PCB Block

typedef struct {

struct {

byte pcb;

byte cid;

byte nad;

} prologue;

struct {

byte size;

byte \*data;

} inf;

} PcbBlock;

// Member variables

TagInfo tag;

/////////////////////////////////////////////////////////////////////////////////////

// Contructors

/////////////////////////////////////////////////////////////////////////////////////

MFRC522Extended() : MFRC522() {};

MFRC522Extended(uint8\_t rst) : MFRC522(rst) {};

MFRC522Extended(uint8\_t ss, uint8\_t rst) : MFRC522(ss, rst) {};

/////////////////////////////////////////////////////////////////////////////////////

// Functions for communicating with PICCs

/////////////////////////////////////////////////////////////////////////////////////

StatusCode PICC\_Select(Uid \*uid, byte validBits = 0) override; // overrride

StatusCode PICC\_RequestATS(Ats \*ats);

StatusCode PICC\_PPS(); // PPS command without bitrate parameter

StatusCode PICC\_PPS(TagBitRates sendBitRate, TagBitRates receiveBitRate); // Different D values

/////////////////////////////////////////////////////////////////////////////////////

// Functions for communicating with ISO/IEC 14433-4 cards

/////////////////////////////////////////////////////////////////////////////////////

StatusCode TCL\_Transceive(PcbBlock \*send, PcbBlock \*back);

StatusCode TCL\_Transceive(TagInfo \* tag, byte \*sendData, byte sendLen, byte \*backData = NULL, byte \*backLen = NULL);

StatusCode TCL\_TransceiveRBlock(TagInfo \*tag, bool ack, byte \*backData = NULL, byte \*backLen = NULL);

StatusCode TCL\_Deselect(TagInfo \*tag);

/////////////////////////////////////////////////////////////////////////////////////

// Support functions

/////////////////////////////////////////////////////////////////////////////////////

static PICC\_Type PICC\_GetType(TagInfo \*tag);

using MFRC522::PICC\_GetType;// // make old PICC\_GetType(byte sak) available, otherwise would be hidden by PICC\_GetType(TagInfo \*tag)

// Support functions for debuging

void PICC\_DumpToSerial(TagInfo \*tag);

using MFRC522::PICC\_DumpToSerial; // make old PICC\_DumpToSerial(Uid \*uid) available, otherwise would be hidden by PICC\_DumpToSerial(TagInfo \*tag)

void PICC\_DumpDetailsToSerial(TagInfo \*tag);

using MFRC522::PICC\_DumpDetailsToSerial; // make old PICC\_DumpDetailsToSerial(Uid \*uid) available, otherwise would be hidden by PICC\_DumpDetailsToSerial(TagInfo \*tag)

void PICC\_DumpISO14443\_4(TagInfo \*tag);

/////////////////////////////////////////////////////////////////////////////////////

// Convenience functions - does not add extra functionality

/////////////////////////////////////////////////////////////////////////////////////

bool PICC\_IsNewCardPresent() override; // overrride

bool PICC\_ReadCardSerial() override; // overrride

};

#endif