CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

PROTOCOLOS DE COMUNICACIÓN EN SISTEMAS EMBEBIDOS

Trabajo final:

Driver para módulo MAX30102 para Oximetría y pulsímetro

Alumno: Lucas Zalazar

Profesores: Gonzalo Sanchez

Pablo Gómez

Año 2021

INTRODUCCIÓN

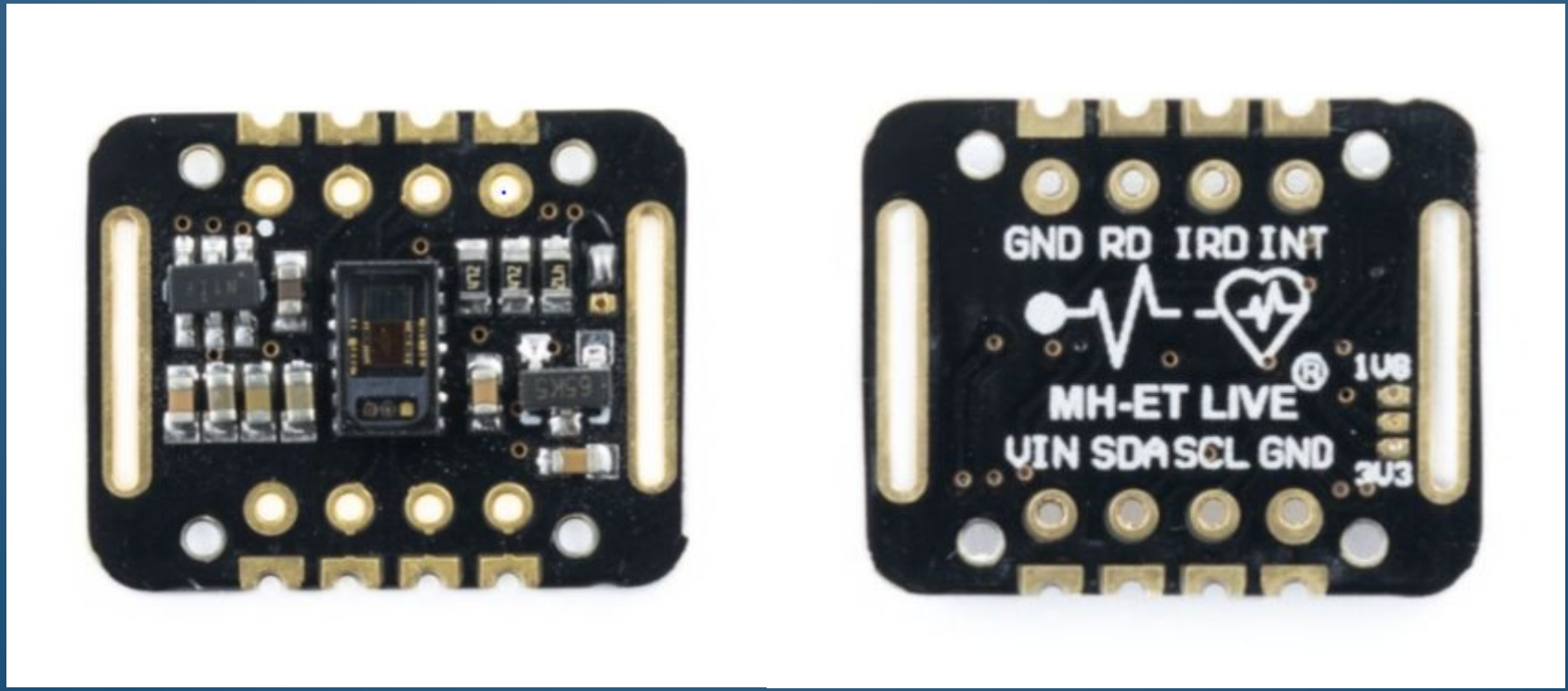
Pulsímetro

Monitor de frecuencia cardíaca.

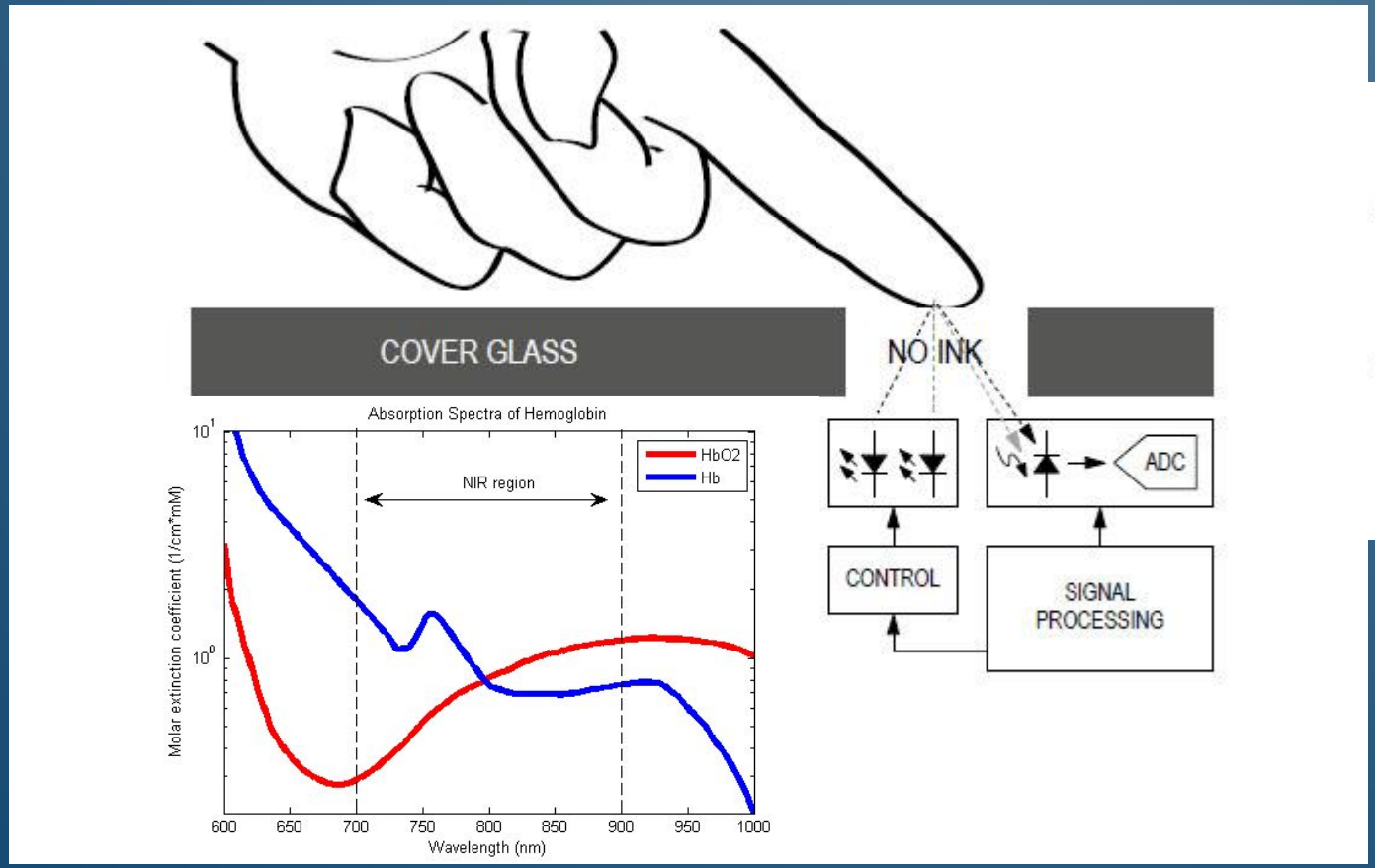
Oximetría

Monitor de saturación de oxígeno en sangre.

MAX 30102



PRINCIPIO DE FUNCIONAMIENTO



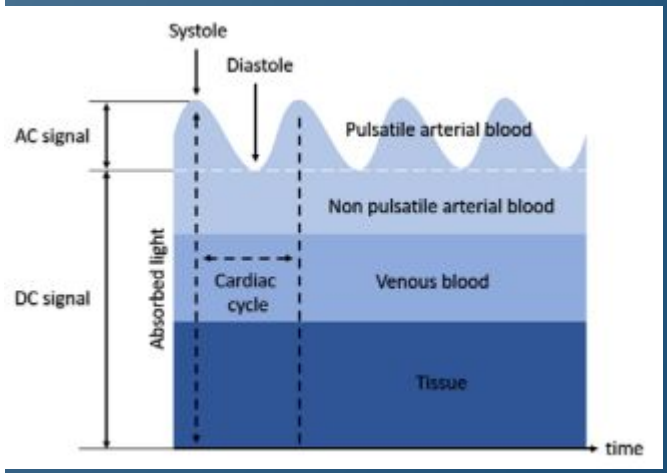
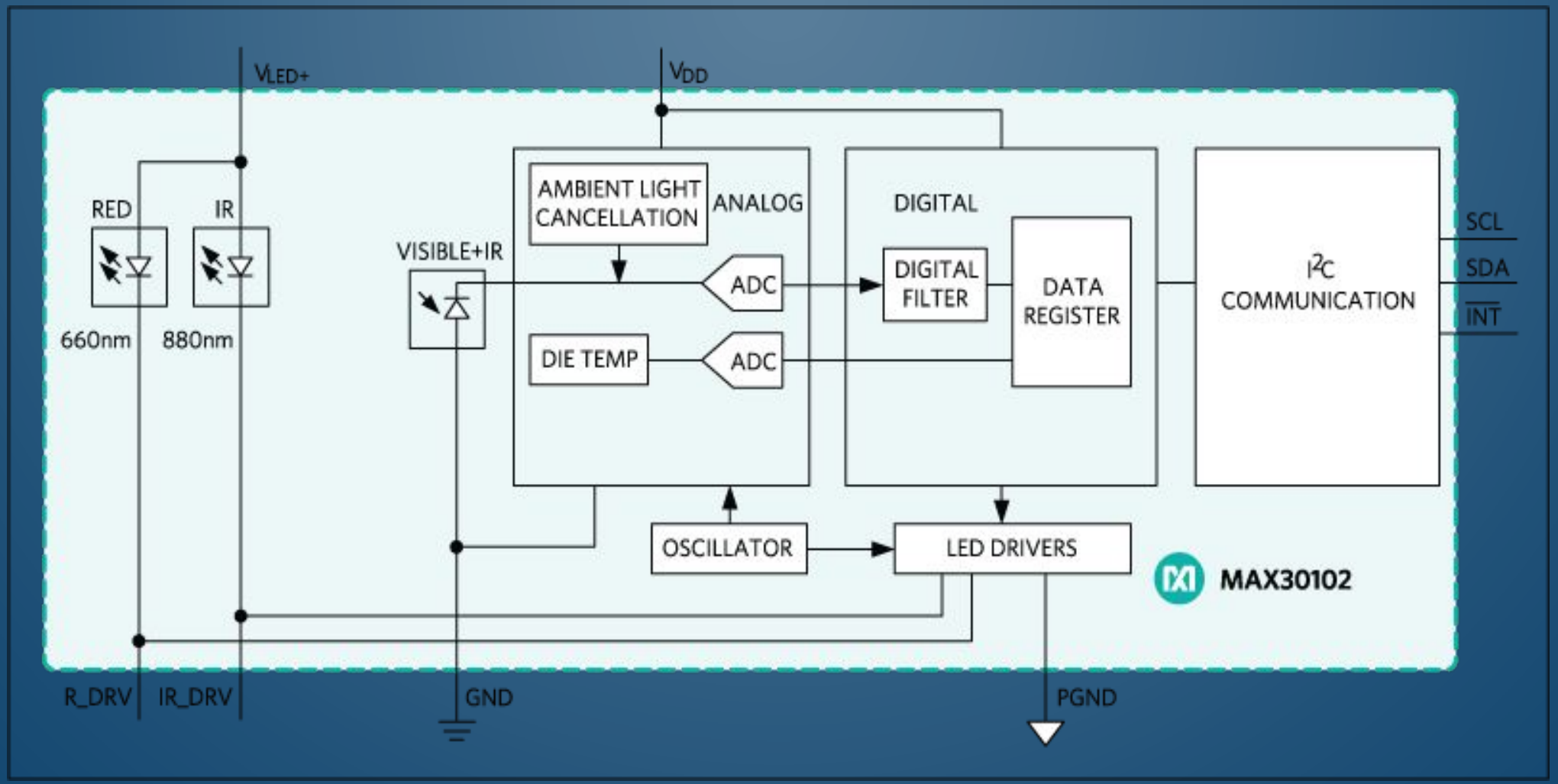


DIAGRAMA FUNCIONAL



REGISTROS

FIFO												
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W	
Overflow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W	
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W	
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W	
CONFIGURATION												
FIFO Configuration	SMP_AVE[2:0]			FIFO_ ROLL OVER_EN	FIFO_A_FULL[3:0]				0x08	0x00	R/W	
Mode Configuration	SHDN	RESET				MODE[2:0]			0x09	0x00	R/W	
SpO_2 Configuration	0 (Reserved)	SPO2_ADC_RGE [1:0]		SPO2_SR[2:0]			LED_PW[1:0]		0x0A	0x00	R/W	
RESERVED									0x0B	0x00	R/W	
LED Pulse Amplitude	LED1_PA[7:0]								0x0C	0x00	R/W	
	LED2_PA[7:0]								0x0D	0x00	R/W	
RESERVED									0x0E	0x00	R/W	
RESERVED									0x0F	0x00	R/W	
Multi-LED Mode Control Registers		SLOT2[2:0]			SLOT1[2:0]				0x11	0x00	R/W	
		SLOT4[2:0]			SLOT3[2:0]				0x12	0x00	R/W	

LECTURA DE DATOS EN FIFO

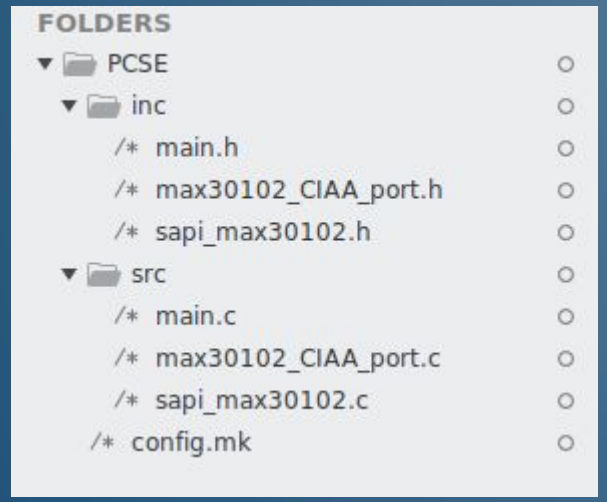
The central processor evaluates the number of samples to be read from the FIFO:

```
NUM_AVAILABLE_SAMPLES = FIFO_WR_PTR - FIFO_RD_PTR
(Note: pointer wrap around should be taken into account)
NUM_SAMPLES_TO_READ = < less than or equal to NUM_AVAILABLE_SAMPLES >
```

Second transaction: Read `NUM_SAMPLES_TO_READ` samples from the FIFO:

```
START;
Send device address + write mode
Send address of FIFO_DATA;
REPEATED_START;
Send device address + read mode
for (i = 0; i < NUM_SAMPLES_TO_READ; i++) {
Read FIFO_DATA;
Save LED1[23:16];
Read FIFO_DATA;
Save LED1[15:8];
Read FIFO_DATA;
Save LED1[7:0];
Read FIFO_DATA;
Save LED2[23:16];
Read FIFO_DATA;
Save LED2[15:8];
Read FIFO_DATA;
Save LED2[7:0];
Read FIFO_DATA;
}
STOP;
START;
Send device address + write mode
Send address of FIFO_RD_PTR;
Write FIFO_RD_PTR;
STOP;
```

LIBRERÍA MAX30102



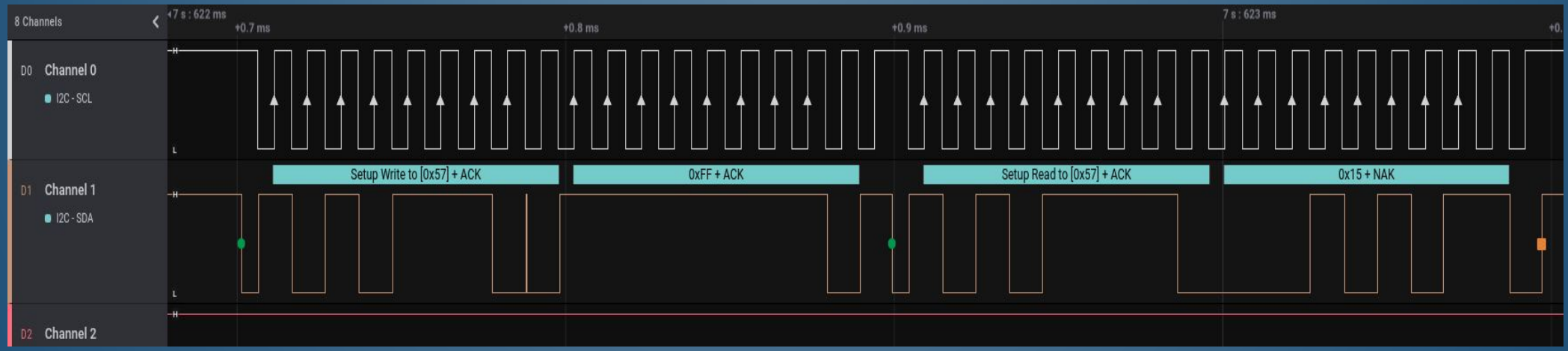

```
/* Mis funciones */
void    initStructMax30102( void );

/* Inicializar device */
bool_t  max30102_Init( max30102_t driver_config );
bool_t  max30102_setup ( max30102_config_t _configDevice );
bool_t  max30102_config ( uint8_t _register, uint8_t _param, uint8_t shitf );
bool_t  max30102_reset  ( void );

/* Lectura de Part ID y Revision ID */
uint8_t max30102_readPartID         ( void );
uint8_t max30102_readRevisionID     ( void );

bool_t  max30102_clearFIFO          ( void );
float_t max30102_readNewValue       ( void );
int16_t max30102_check              ( void );
uint8_t max30102_getWritePointer    ( void );
uint8_t max30102_getReadPointer     ( void );

float_t max30102_oxygenSaturation   (uint32_t * ledIr, uint32_t * ledR, int32_t numSamples );
uint32_t max30102_hearBeat          ( void );

/* Funcion para enmascarar registros */
void    max30102_maskRegister ( uint8_t _register, uint8_t _mask, uint8_t bitMask );
```

```
bool_t i2cInit_CIAA_port ( i2cMap_t i2cNumber, uint32_t clockRateHz );
bool_t max30102Write_CIAA_port ( i2cMap_t  i2cNumber, uint8_t  i2cSlaveAddress, uint8_t registerAddr, uint8_t data );
bool_t max30102Read_CIAA_port ( i2cMap_t  i2cNumber, uint8_t  i2cSlaveAddress, uint8_t registerAddr,
                                uint8_t* receiveDataBuffer, uint16_t receiveDataBufferSize );
uint32_t delay_CIAA_port ( void );
```

DEMOSTRACIÓN FUNCIONAL

VIDEO

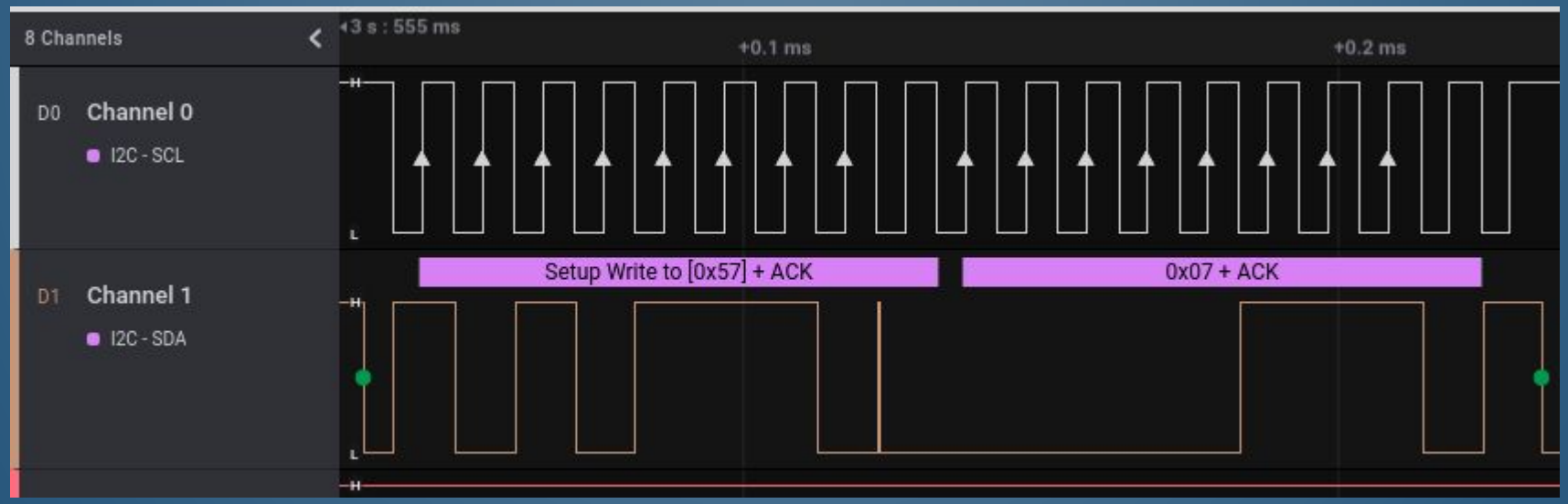
LECTURA/ESCRITURA DE REGISTROS

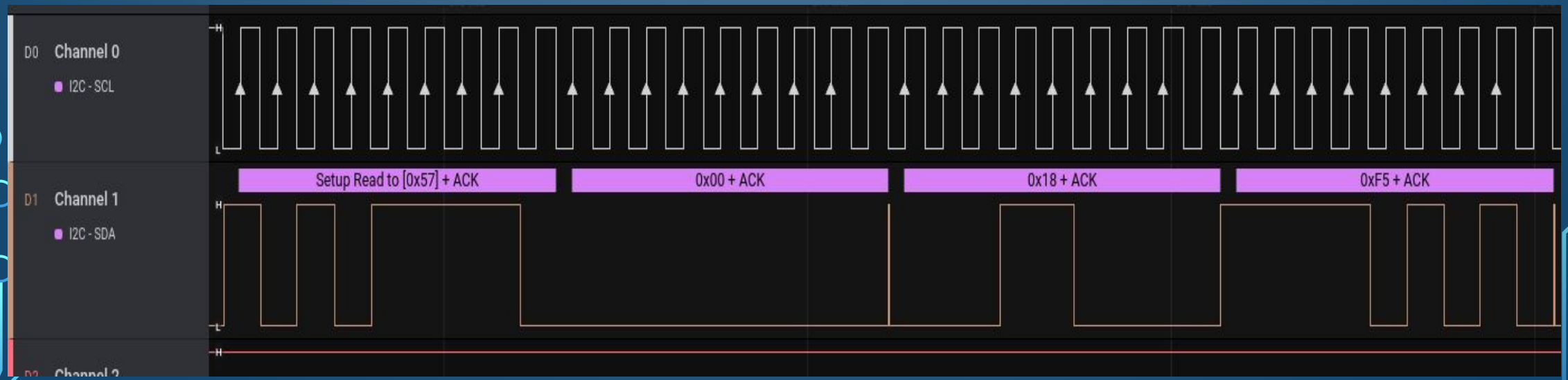


PART ID				
Revision ID	REV_ID[7:0]	0xFE	0xXX*	R
Part ID	PART_ID[7]	0xFF	0x15	R

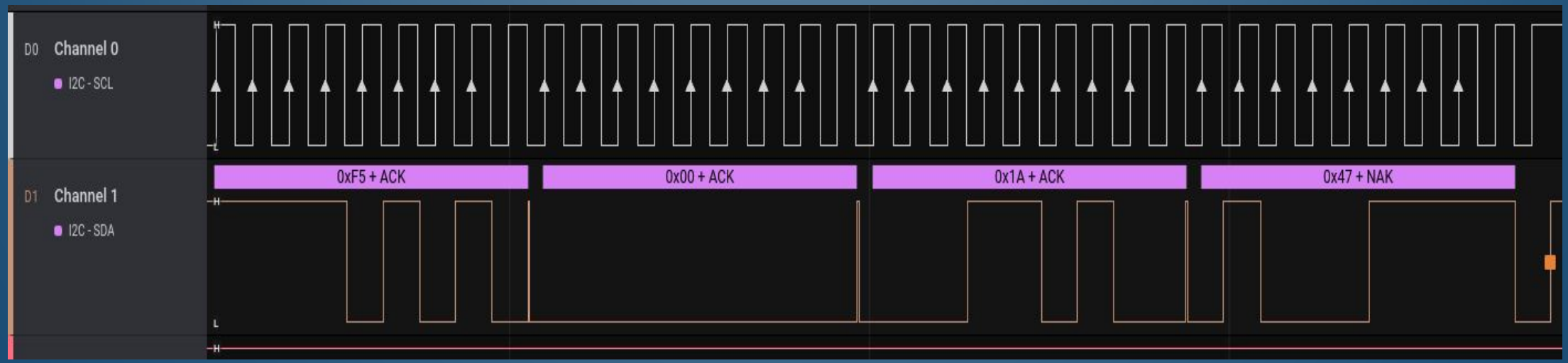
**XX denotes a 2-digit hexadecimal number (00 to FF) for part revision identification. Contact Maxim Integrated for the revision ID number assigned for your product.*

LECTURA DE DATOS





LECTURA DE DATOS II



PREGUNTAS

¡Gracias!