

[ANEXO I]

[Código fuente del archivo SGR765.icd]

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL">
  <Header id="" nameStructure="IEDName">
    </Header>
  <Communication>
    <SubNetwork name="subnetwork1" type="8-MMS">
      <Text>Station bus</Text>
      <BitRate unit="b/s">10</BitRate>
      <ConnectedAP iedName="simpleIO" apName="accessPoint1">
        <Address>
          <P type="IP">10.0.0.2</P>
          <P type="IP-SUBNET">255.255.255.0</P>
          <P type="IP-GATEWAY">10.0.0.1</P>
          <P type="OSI-TSEL">0001</P>
          <P type="OSI-PSEL">00000001</P>
          <P type="OSI-SSEL">0001</P>
        </Address>
        <GSE ldInst="GenericIO" cbName="gcbEvents">
          <Address>
            <P type="VLAN-ID">1</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="MAC-Address">01-0c-cd-01-00-01</P>
            <P type="APPID">1000</P>
          </Address>
        </GSE>
        <GSE ldInst="GenericIO" cbName="gcbAnalogValues">
          <Address>
            <P type="VLAN-ID">1</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="MAC-Address">01-0c-cd-01-00-01</P>
            <P type="APPID">1000</P>
          </Address>
        </GSE>
      </ConnectedAP>
    </SubNetwork>
  </Communication>
  <IED name="simpleIO">
    <Services>
      <DynAssociation />
      <GetDirectory />
      <GetDataObjectDefinition />
      <GetDataSetValue />
      <DataSetDirectory />
      <ReadWrite />
      <GetCBValues />
      <ConflNs fixPrefix="true" fixLnInst="true" />
      <GOOSE max="5" />
      <GSSE max="5" />
      <FileHandling />
      <GSEDir />
      <TimerActivatedControl />
    </Services>
    <AccessPoint name="accessPoint1">
```

```

    <Server>
    <Authentication />
    <LDevice inst="GenericIO">
    <LN0 lnClass="LLN0" lnType="LLN01" inst="">
    <GSEControl appID="events" name="gcbEvents" type="GOOSE" dataSet="Events"
confRev="2" minTime="10" maxTime="3000" />
    <GSEControl appID="analog" name="Voltaje" type="GOOSE"
datSet="AnalogValues" confRev="2"/>
    <DataSet name="Events" desc="Events">
    <FCDA ldInst="GenericIO" lnClass="GGIO" fc="ST" lnInst="1" doName="SPCS01"
daName="stVal" />
    </DataSet>
    <DataSet name="AnalogValues" desc="analog values">
    <FCDA ldInst="GenericIO" lnClass="GGIO" fc="MX" lnInst="1" doName="A" />
    </DataSet>
    </LN0>
    <LN lnClass="GGIO" lnType="GGIO1" inst="1" prefix="">
    <DOI name="SPCS01">
    <DAI name="ctlModel">
    <Val>direct-with-normal-security</Val>
    </DAI>
    </DOI>
    </LN>
    </LDevice>
    </Server>
    </AccessPoint>
    </IED>
    <DataTypeTemplates>GGIO1
    <LNNodeType id="LLN01" lnClass="LLN0">
    <DO name="Health" type="INS_1_Beh" />
    </LNNodeType>
    <LNNodeType id="GGIO1" lnClass="GGIO">
    <DO name="SPCS01" type="SP1" />
    <DO name="Pos" type="POS" />
    <DO name="A" type="A" />
    </LNNodeType>
    <DOType id="POS" cdc="SPC">
    <DA name="stVal" bType="BOOLEAN" fc="ST" />
    <DA name="ctlVal" bType="BOOLEAN" fc="CO" />
    <DA name="t" bType="Timestamp" fc="MX" />
    </DOType>
    <DOType id="A" cdc="MV">
    <DA name="fhA" bType="FLOAT32" fc="MX" />
    <DA name="fhB" bType="FLOAT32" fc="MX" />
    <DA name="fhC" bType="FLOAT32" fc="MX" />
    <DA name="fhN" bType="FLOAT32" fc="MX" />
    <DA name="Pot" bType="FLOAT32" fc="MX" />
    </DOType>
    <DOType id="SP1" cdc="SPC">
    <DA name="stVal" bType="BOOLEAN" fc="ST" dchg="true" />
    <DA name="ctlModel" type="CtlModels" bType="Enum" fc="CF" />
    <DA name="ctlVal" bType="BOOLEAN" fc="CO" />
    <DA name="t" bType="Timestamp" fc="ST" />
    </DOType>
    <DOType id="INC_1_Mod" cdc="INC">
    <DA name="q" bType="Quality" fc="ST" qchg="true" />
    <DA name="t" bType="Timestamp" fc="ST" />
    <DA name="ctlModel" type="CtlModels" bType="Enum" fc="CF" />
    </DOType>
    <DOType id="INS_1_Beh" cdc="INS">
    <DA name="stVal" bType="INT32" fc="ST" dchg="true" />

```

```
<DA name="q" bType="Quality" fc="ST" qchg="true" />
<DA name="t" bType="Timestamp" fc="ST" />
</DOType>
<DOType id="LPL_1_NamPlt" cdc="LPL">
  <DA name="vendor" bType="VisString255" fc="DC" />
  <DA name="swRev" bType="VisString255" fc="DC" />
  <DA name="d" bType="VisString255" fc="DC" />
  <DA name="configRev" bType="VisString255" fc="DC" />
  <DA name="ldNs" bType="VisString255" fc="EX" />
</DOType>
<DOType id="MV_1_AnIn1" cdc="MV">
  <DA name="mag" type="AnalogueValue_1" bType="Struct" fc="MX" dchg="true" />
  <DA name="q" bType="Quality" fc="MX" qchg="true" />
  <DA name="t" bType="Timestamp" fc="MX" />
</DOType>
<DAType id="AnalogueValue_1">
  <BDA name="f" bType="FLOAT32" />
</DAType>
<EnumType id="CtlModels">
  <EnumVal ord="0">status-only</EnumVal>
  <EnumVal ord="1">direct-with-normal-security</EnumVal>
  <EnumVal ord="2">sbo-with-normal-security</EnumVal>
  <EnumVal ord="3">direct-with-enhanced-security</EnumVal>
  <EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
</EnumType>
</DataTypeTemplates>
</SCL>
```

[ANEXO II]

[Archivo de configuración para compilación Makefile]

```
LIBIEC_HOME=../..
PROJECT_BINARY_NAME = ServidorGOOSE
PROJECT_SOURCES = servidorGOOSE.c
PROJECT_SOURCES += static_model.c
PROJECT_ICD_FILE =SGR765.icd
include $(LIBIEC_HOME)/make/target_system.mk
include $(LIBIEC_HOME)/make/stack_includes.mk
all: $(PROJECT_BINARY_NAME)
include $(LIBIEC_HOME)/make/common_targets.mk
model: $(PROJECT_ICD_FILE)
    java -jar $(LIBIEC_HOME)/tools/model_generator/genmodel.jar $
(PROJECT_ICD_FILE)
$(PROJECT_BINARY_NAME): $(PROJECT_SOURCES) $(LIB_NAME)
$(CC) $(CFLAGS) $(LDFLAGS) -o $(PROJECT_BINARY_NAME) $(PROJECT_SOURCES) $
(INCLUDES) $(LIB_NAME) $(LDLIBS)
clean:
    rm -f $(PROJECT_BINARY_NAME)
    cp $(PROJECT_ICD_FILE) $(LIBIEC_HOME)/tools/model_generator/$
(PROJECT_ICD_FILE)
    sleep 1
    pwd
    java -jar $(LIBIEC_HOME)/tools/model_generator/genmodel.jar $
(PROJECT_ICD_FILE)
    sleep 1
```

[ANEXO III]

[Código fuente del servidor GOOSE]

```
servidorGOOSE.c
#include "IEC 61850_server.h"
#include "hal_thread.h"
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include "static_model.h"
/* importa un modelo IEC 61850 creado desde un archivo SCL */
extern IedModel iedModel;
static int running = 0;
static IedServer iedServer = NULL;
void traza(float d)
{
    /* Se escriben los valores de potencia creciente aleatoria
    *en el archivo datos.dat
    */
    FILE *f;
    f = fopen("/var/www/htdocs/PTRweb/datos.dat", "w+");
    fprintf(f, "%f\n", d);
    fclose(f);
}
int linea_activa()
{
    /* Se escriben el estado de linea SGR-ARE en el archivo activa.dat */
    FILE *fp;
    char buffer[10];
    fp = fopen ( "/var/www/htdocs/PTRweb/activa.dat", "r" );
    fscanf(fp, "%s", buffer);
    printf("%s",buffer) ;
    if (buffer[0]=='1') return 1;
    else return 0;
    fclose ( fp );
}
void sigint_handler(int signalId)
{
    running = 0;
}
int main(int argc, char** argv) {
    iedServer = IedServer_create(&iedModel);
    if (argc > 1) {
        char* ethernetIfcID = argv[1];
        printf("Usando interfaz GOOSE: %s\n", ethernetIfcID);
        IedServer_setGooseInterfaceId(iedServer, ethernetIfcID);
    }
    /* Servidor MMS corriendo en el puerto 102 */
    IedServer_start(iedServer, 102);
    if (!IedServer_isRunning(iedServer)) {
        printf("Starting server failed! Exit.\n");
        IedServer_destroy(iedServer);
        exit(-1);
    }
    /* Publicador GOOSE */
    IedServer_enableGoosePublishing(iedServer);
    running = 1;
    /* Variables con los voltajes de fase y valores de potencia */
    float fhA = 0.f;
    float fhB = 0.f;
    float fhC = 0.f;
}
```

```

float fhN = 0.f;
float Pot = 0.f;
Pot=1500;
while (running) {
    fhA=rand() % 6 + 750;
    fhB=rand() % 6 + 750;
    fhC=rand() % 6 + 750;
    fhN=rand() % 6;
    Pot=rand() % 50 + Pot;
    IedServer_lockDataModel(iedServer);
    /* Publicacion de los valores de voltaje y potencia*/
    IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhA, fhA);
    IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhB, fhB);
    IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhC, fhC);
    IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_fhN, fhN);
    IedServer_updateFloatAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_A_Pot, Pot);
    IedServer_unlockDataModel(iedServer);
    if (linea_activa()==1){
        traza(Pot);
    }else{
        traza(0);
        Pot=1500;
    }
    Thread_sleep(1000);
    if (Pot>2000) {
/* Si se supera el limite de transferencia de potencia supera los
2000MW, la
* linea SGR-ARE dispara y se envia un mensaje GOOSE de alta prioridad
para
* activar los botes de carga para el codigo C1
*/
        IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal,0);
        IedServer_updateUTCTimeAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_t, Hal_getTimeInMs());
        Pot=0;
    }
    if (linea_activa()==1 && Pot<2000) {
/* Si se supera el limite de transferencia de potencia supera los 2000MW, la
linea
* SGR-ARE dispara y se envia un mensaje GOOSE de alta prioridad para activar
los
* botes de carga para el codigo C1
*/
        IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal,1);
        IedServer_updateUTCTimeAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_t, Hal_getTimeInMs());
    }
    if (linea_activa()==0 ){
        /* Linea inactiva
        */
        IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal,0);
        IedServer_updateUTCTimeAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_t, Hal_getTimeInMs());

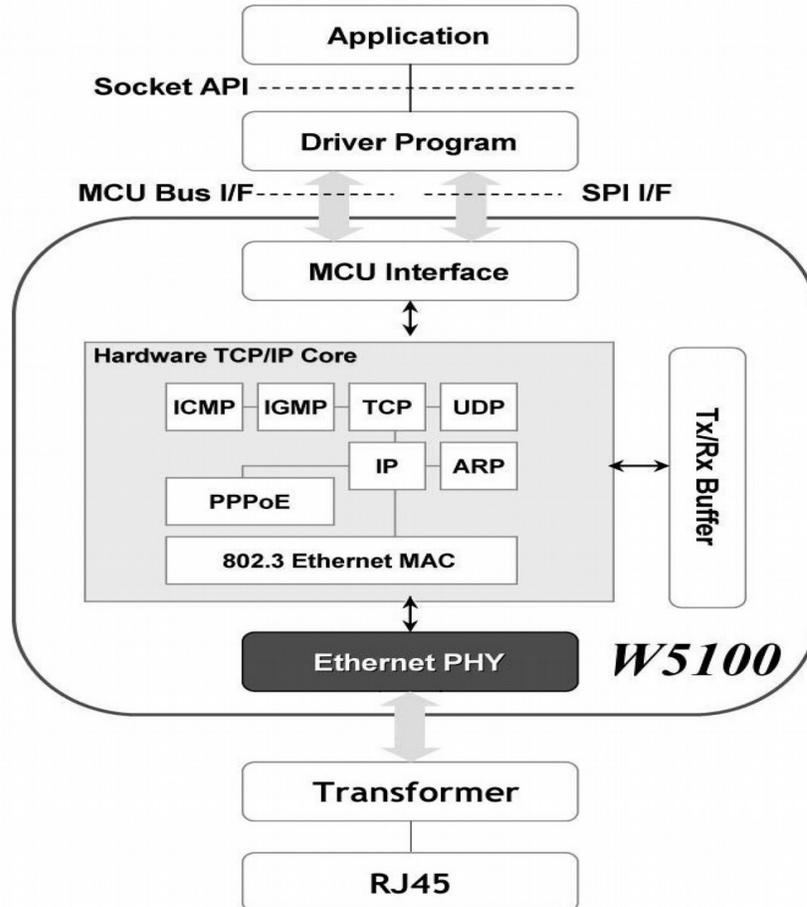
```

```
    }  
}  
  
/* Detiene el servidor MMS y cierra el socket TCP */  
IedServer_stop(iedServer);  
  
/* Libera todos los recursos */  
IedServer_destroy(iedServer);  
}
```

[ANEXO IV]

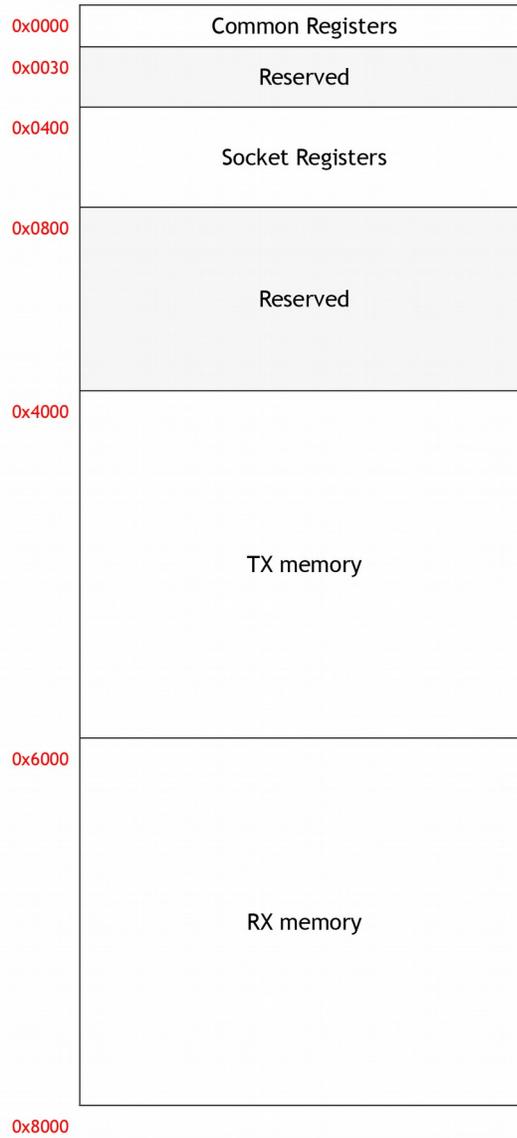
[Especificaciones técnicas del chip W5100]

Block Diagram



2. Memory Map

W5100 is composed of Common Register, Socket Register, TX Memory, and RX Memory as shown below.



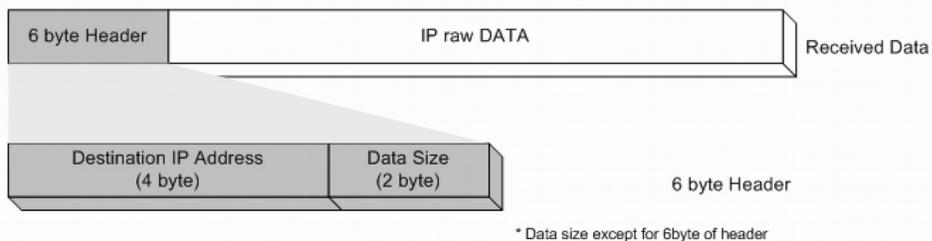
Memory Map

- Received DATA?

It is same as UDP. Refer to “5.2.2 UDP”.

- Receiving Process

This is same as UDP. Refer to “5.2.2 UDP” except the header information and header size.
In case of IP raw, 6byte header is attached to the data received. The header structure is as below.



- Send DATA? / Sending Process

This is same as UDP. Refer to “5.2.2 UDP” except that remote_port information is not needed.

- Complete Sending
- Timeout
- Finished? / Socket Closed

Next actions are same as UDP. Refer to “5.2.2 UDP”.

5.2.4 MAC raw

MAC Raw mode(only supported in socket 0) can be utilized.

- Socket Initialization

It initializes the socket as MAC raw.

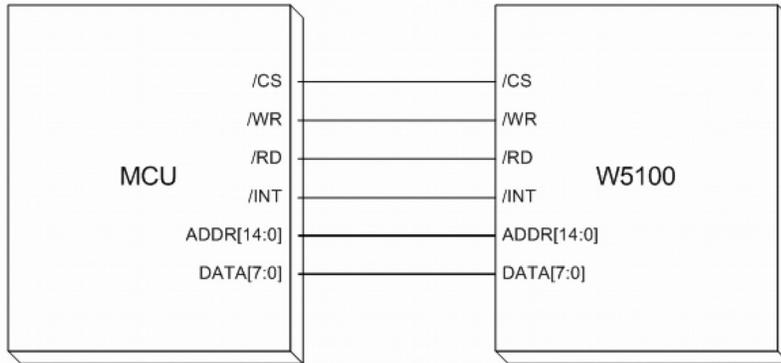
```
{
START:
/* sets MAC raw mode */
```

6. Application Information

For the communication with MCU, W5100 provides Direct, Indirect Bus I/F, and SPI I/F modes.
 For the communication with Ethernet PHY, MII is used.

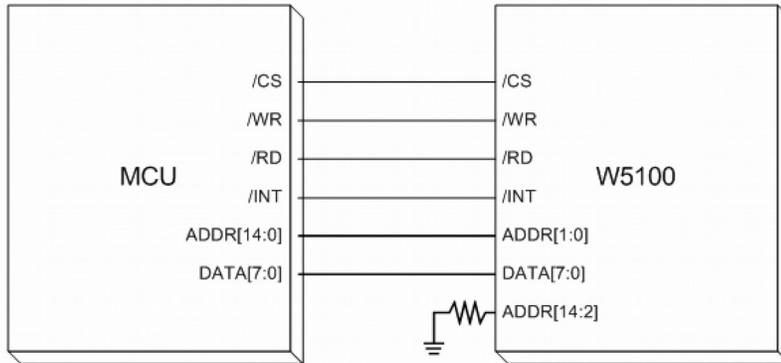
6.1 Direct Bus Interface mode

Direct Bus I/F mode uses 15bit address line and 8bit data line, /CS, /RD, /WR, /INT.



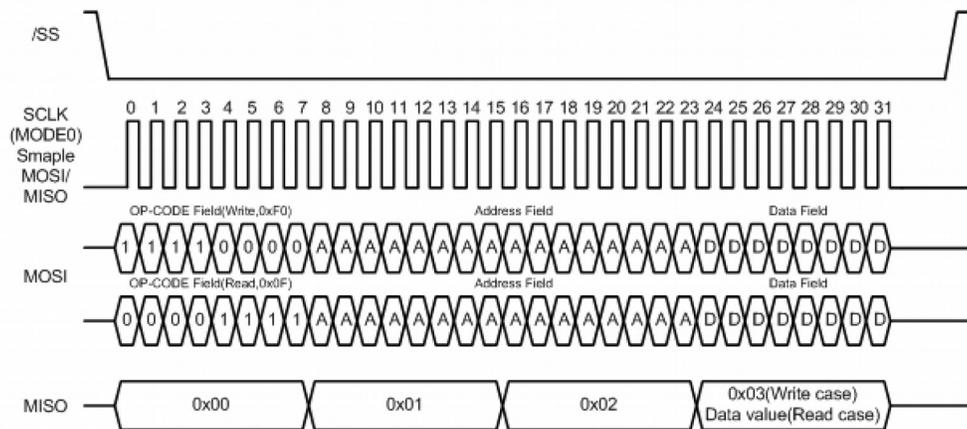
6.2 Indirect Bus Interface mode

Indirect Bus I/F mode uses 2bit address line and 8bit data line, /CS, /RD, /WR, /INT.
 [14:2], other address lines should process Pull-down.



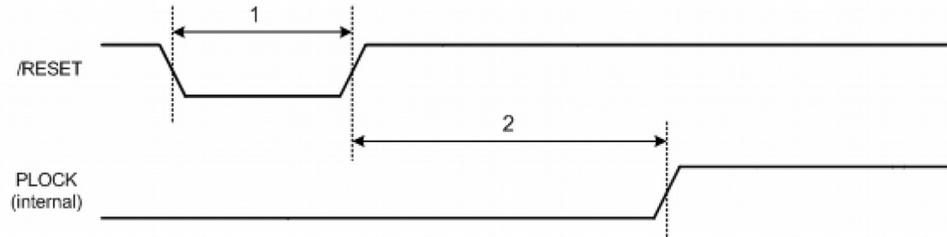
Indirect bus I/F mode related register is as below.

1. Configure Input/Output direction on SPI Master device pins.
 - * /SS (Slave Select) : Output pin
 - * SCLK (Serial Clock) : Output pin
 - * MOSI (Master Out Slave In) : Output pin
 - * MISO (Master In Slave Out) : Input pin
2. Configure /SS as 'High'
3. Configure the registers on SPI Master device.
 - * SPI Enable bit on SPCR register (SPI Control Register)
 - * Master/Slave select bit on SPCR register
 - * SPI Mode bit on SPCR register
 - * SPI data rate bit on SPCR register and SPSR register (SPI State Register)
4. Write desired value for transmission on SPDR register (SPI Data Register).
5. Configure /SS as 'Low' (data transfer start)
6. Wait for reception complete
7. If all data transmission ends, configure /SS as 'High'



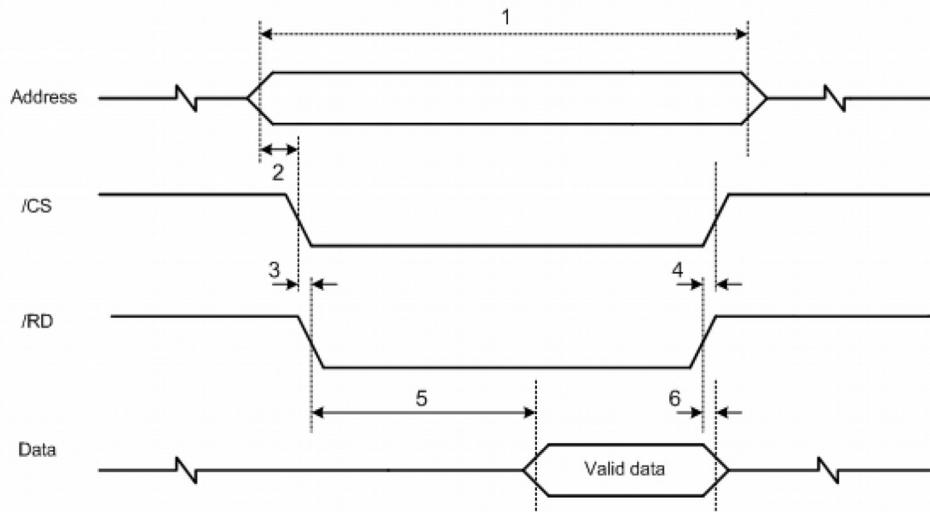
AC Characteristics

Reset Timing



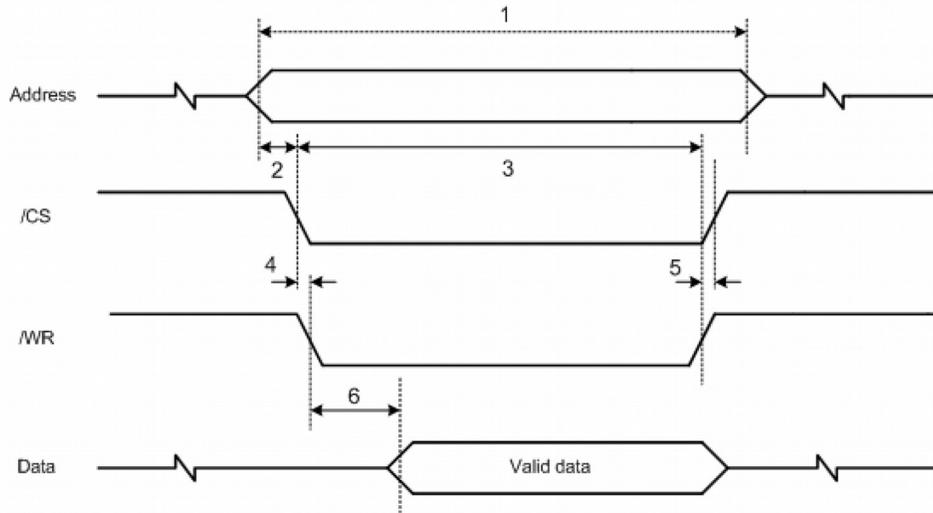
Description		Min	Max
1	Reset Cycle Time	2 us	-
2	/RESET to internal PLOCK	-	10 ms

Register/Memory READ Timing



Description		Min	Max
1	Read Cycle Time	80 ns	-
2	Valid Address to /CS low time	8 ns	-
3	/CS low to /RD low time	-	1 ns
4	/RD high to /CS high time	-	1 ns
5	/RD low to Valid Data Output time	-	80 ns
6	/RD high to Data High-Z Output time	-	1 ns

Register/Memory WRITE Timing



Description		Min	Max
1	Write Cycle Time	70 ns	-
2	Valid Address to /CS low time	7 ns	-
3	/CS low to /WR high time	70 ns	-
4	/CS low to /WR low time	-	1 ns
5	/WR high to /CS high time	-	1 ns
6	/WR low to Valid Data time	-	14 ns

Crystal Characteristics

Parameter	Range
Frequency	25 MHz
Frequency Tolerance (at 25°C)	±30 ppm
Shunt Capacitance	7pF Max
Drive Level	100uW
Load Capacitance	27pF
Aging (at 25°C)	±3ppm / year Max

Transformer Characteristics

Parameter	Transmit End	Receive End
Turn Ratio	1:1	1:1
Inductance	350 uH	350 uH

Symmetrical TX & RX channels for auto MDI/MDIX capability

8. IR Reflow Temperature Profile (Lead-Free)

Moisture Sensitivity Level : 3

Dry Pack Required : Yes

Average Ramp-Up Rate ($T_{S_{max}}$ to T_p)	3° C/second max.
Preheat <ul style="list-style-type: none"> - Temperature Min ($T_{S_{min}}$) - Temperature Max ($T_{S_{max}}$) - Time ($t_{S_{min}}$ to $t_{S_{max}}$) 	150 °C 200 °C 60-180 seconds
Time maintained above: <ul style="list-style-type: none"> - Temperature (T_L) - Time (t_L) 	217 °C 60-150 seconds
Peak/Classification Temperature (T_p)	260 + 0 °C
Time within 5 °C of actual Peak Temperature (t_p)	20-40 seconds
Ramp-Down Rate	6 °C/second max.
Time 25 °C to Peak Temperature	8 minutes max.

