AURIX_ETH 说明文档

-曾喆 13041230620

-北京晶川电子

1. 文档说明：

本文档讲解的是 infineon 的 AURIX 的 TC297 单片机的 ETH 模块的 MCAL 配置；

2. 软件说明：

MCAL 配置软件：EB Tresos 16；

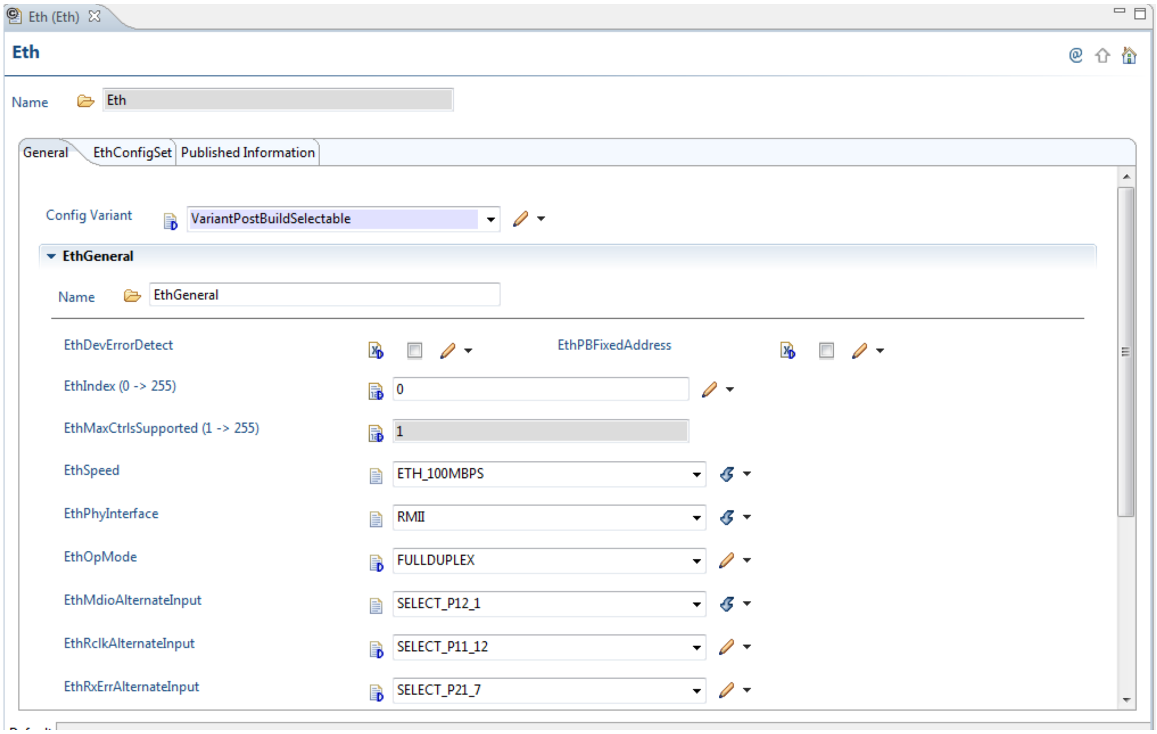
MCAL 软件包：MC-ISAR_AS4XX_AURIX_TC29X_BB_PB_BASE_V500

MC-ISAR_AS4XX_AURIX_TC29X_BB_PB_COM_ENHANCED_V400；

编译软件：Hightec4.6.6；

3. EB Tresos MCAL 配置界面：

创建 EB Tresos 工程，添加 ETH 模块



Name: Eth

Config Variant: VariantPostBuildSelectable

Name: EthGeneral

EthDevErrorDetect □

EthPBFixedAddress □

EthIndex (0 -> 255): 0

EthMaxCtrlsSupported (1 -> 255): 1

EthSpeed: ETH_100MBPS

EthPhyInterface: RMII

EthOpMode: FULLDUPLEX

EthMdioAlternateInput: SELECT_P12_1

EthRclkAlternateInput: SELECT_P11_12

EthRxErrAlternateInput: SELECT_P21_7

EthIndex：不需要修改；

ETHSpeed：客户按照需求设定；

EthPhyInterface：TC297 只支持 RMII 和 MII，客户按照需求设定；

EthOpMode：客户按照需求设定；

EthMidoAlternateInput、EthRclkAlternateInput、EthRxErrAlternateInput 按照实际电路选择；

TC297 可配置的 IO 引脚选项：

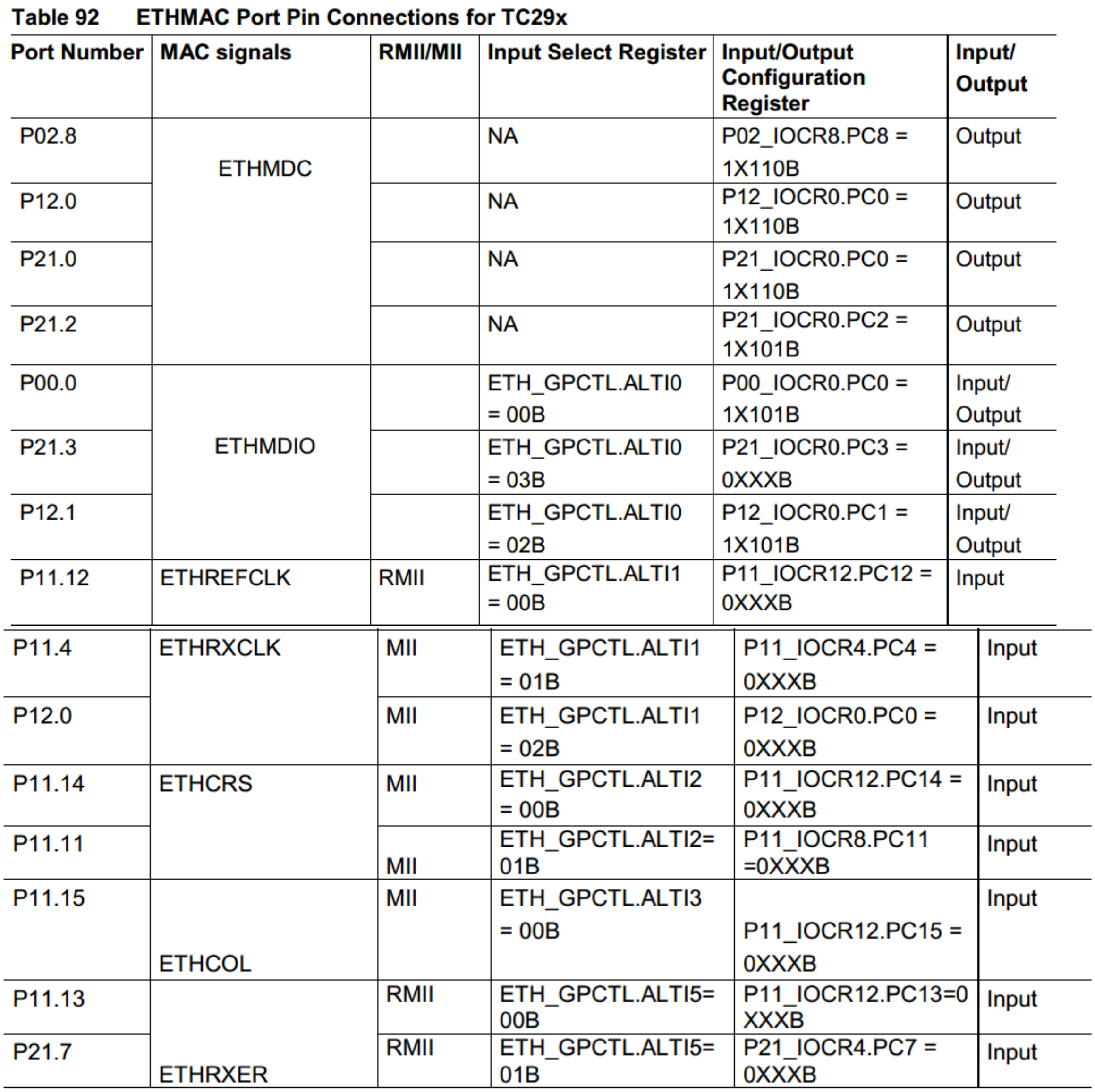
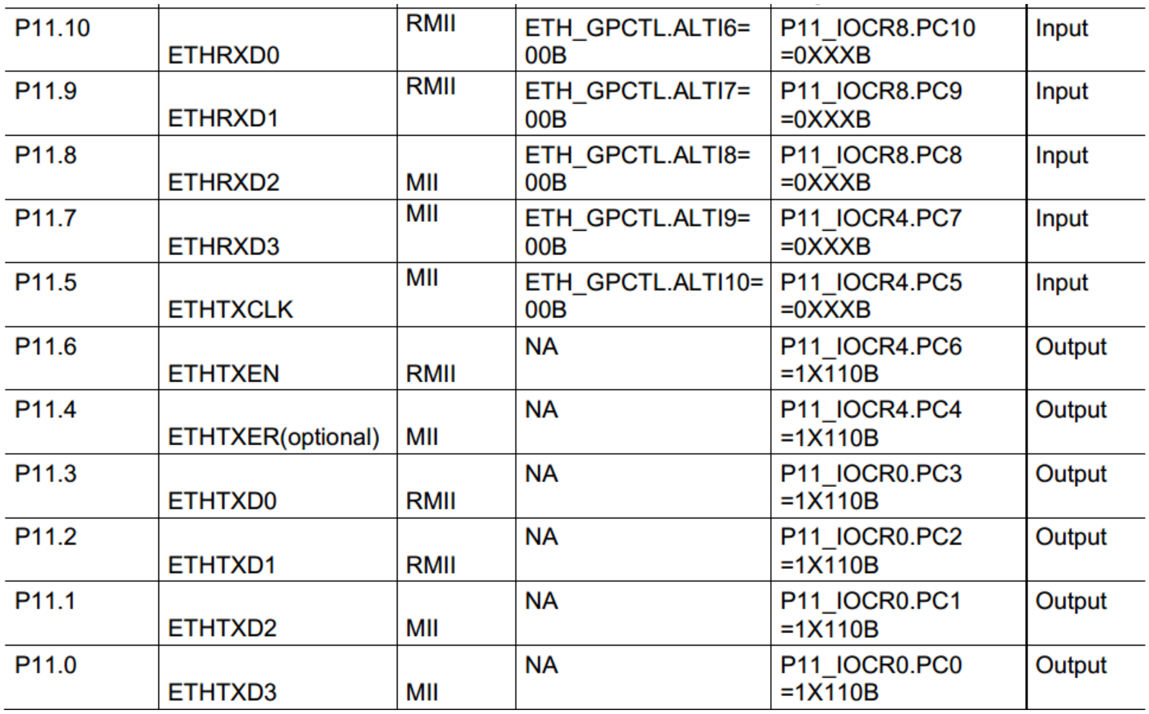


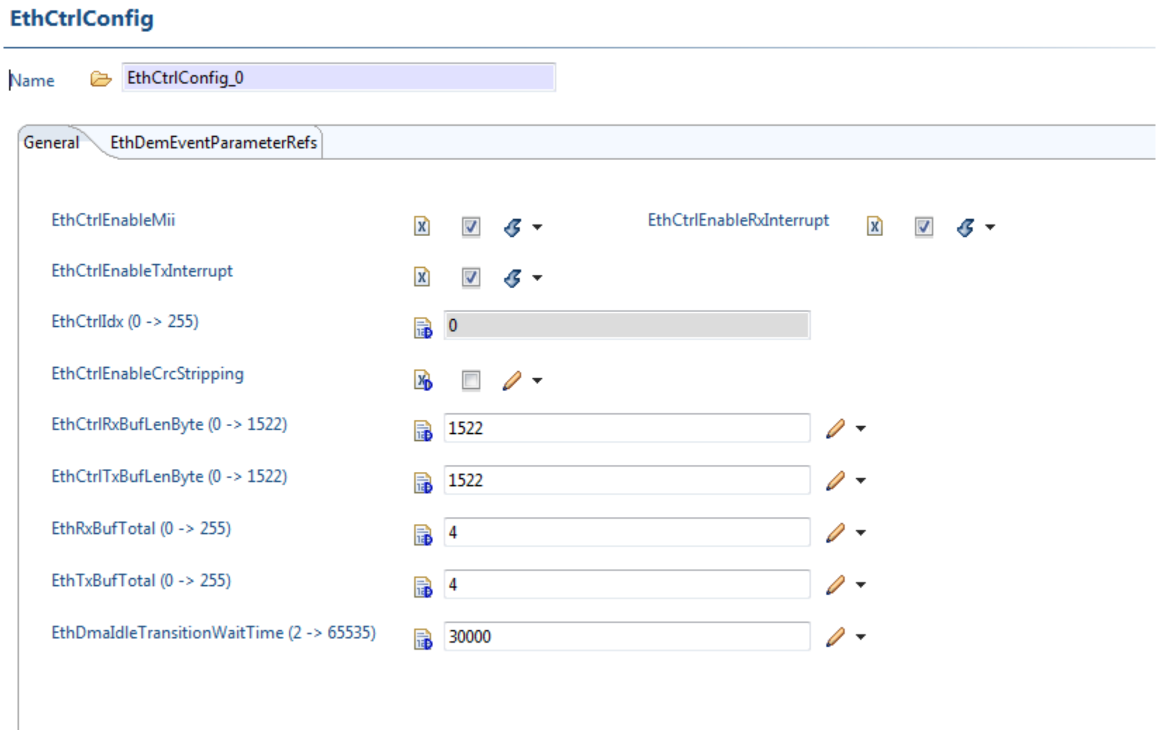
Table 92 ETHMAC Port Pin Connections for TC29x

Port Number	MAC signals	RMII/MII	Input Select Register	Input/Output Configuration Register	Input/ Output
P02.8	ETHMDC		NA	P02_IOCR8.PC8 = 1X110B	Output
P12.0			NA	P12_IOCR0.PC0 = 1X110B	Output
P21.0			NA	P21_IOCR0.PC0 = 1X110B	Output
P21.2			NA	P21_IOCR0.PC2 = 1X101B	Output
P00.0	ETHMDIO		ETH_GPCTL.ALTI0 = 00B	P00_IOCR0.PC0 = 1X101B	Input/ Output
P21.3			ETH_GPCTL.ALTI0 = 03B	P21_IOCR0.PC3 = 0XXXB	Input/ Output
P12.1			ETH_GPCTL.ALTI0 = 02B	P12_IOCR0.PC1 = 1X101B	Input/ Output
P11.12	ETHREFCLK	RMII	ETH_GPCTL.ALTI1 = 00B	P11_IOCR12.PC12 = 0XXXB	Input
P11.4	ETHRXCLK	MII	ETH_GPCTL.ALTI1 = 01B	P11_IOCR4.PC4 = 0XXXB	Input
P12.0		MII	ETH_GPCTL.ALTI1 = 02B	P12_IOCR0.PC0 = 0XXXB	Input
P11.14	ETHCRS	MII	ETH_GPCTL.ALTI2 = 00B	P11_IOCR12.PC14 = 0XXXB	Input
P11.11		MII	ETH_GPCTL.ALTI2= 01B	P11_IOCR8.PC11 =0XXXB	Input
P11.15	ETHCOL	MII	ETH_GPCTL.ALTI3 = 00B	P11_IOCR12.PC15 = 0XXXB	Input
P11.13	ETHRXER	RMII	ETH_GPCTL.ALTI5= 00B	P11_IOCR12.PC13=0XXXB	Input
P21.7		RMII	ETH_GPCTL.ALTI5= 01B	P21_IOCR4.PC7 = 0XXXB	Input



P11.10	ETHRXD0	RMII	ETH_GPCTL.ALTI6=00B	P11_IOCR8.PC10=0XXXB	Input
P11.9	ETHRXD1	RMII	ETH_GPCTL.ALTI7=00B	P11_IOCR8.PC9=0XXXB	Input
P11.8	ETHRXD2	MII	ETH_GPCTL.ALTI8=00B	P11_IOCR8.PC8=0XXXB	Input
P11.7	ETHRXD3	MII	ETH_GPCTL.ALTI9=00B	P11_IOCR4.PC7=0XXXB	Input
P11.5	ETHTXCLK	MII	ETH_GPCTL.ALTI10=00B	P11_IOCR4.PC5=0XXXB	Input
P11.6	ETHTXEN	RMII	NA	P11_IOCR4.PC6=1X110B	Output
P11.4	ETHTXER(optional)	MII	NA	P11_IOCR4.PC4=1X110B	Output
P11.3	ETHTXD0	RMII	NA	P11_IOCR0.PC3=1X110B	Output
P11.2	ETHTXD1	RMII	NA	P11_IOCR0.PC2=1X110B	Output
P11.1	ETHTXD2	MII	NA	P11_IOCR0.PC1=1X110B	Output
P11.0	ETHTXD3	MII	NA	P11_IOCR0.PC0=1X110B	Output

点击 EthConfigSet->EthConfigSet_0->EthCtrlConfig_0



EthCtrlConfig

Name EthCtrlConfig_0

EthCtrlEnableMii	☑	EthCtrlEnableRxInterrupt	☑
EthCtrlEnableTxInterrupt	☑		
EthCtrlIdx (0 -> 255)	0		
EthCtrlEnableCrcStripping	□		
EthCtrlRxBufLenByte (0 -> 1522)	1522		
EthCtrlTxBufLenByte (0 -> 1522)	1522		
EthRxBufTotal (0 -> 255)	4		
EthTxBufTotal (0 -> 255)	4		
EthDmaIdleTransitionWaitTime (2 -> 65535)	30000		

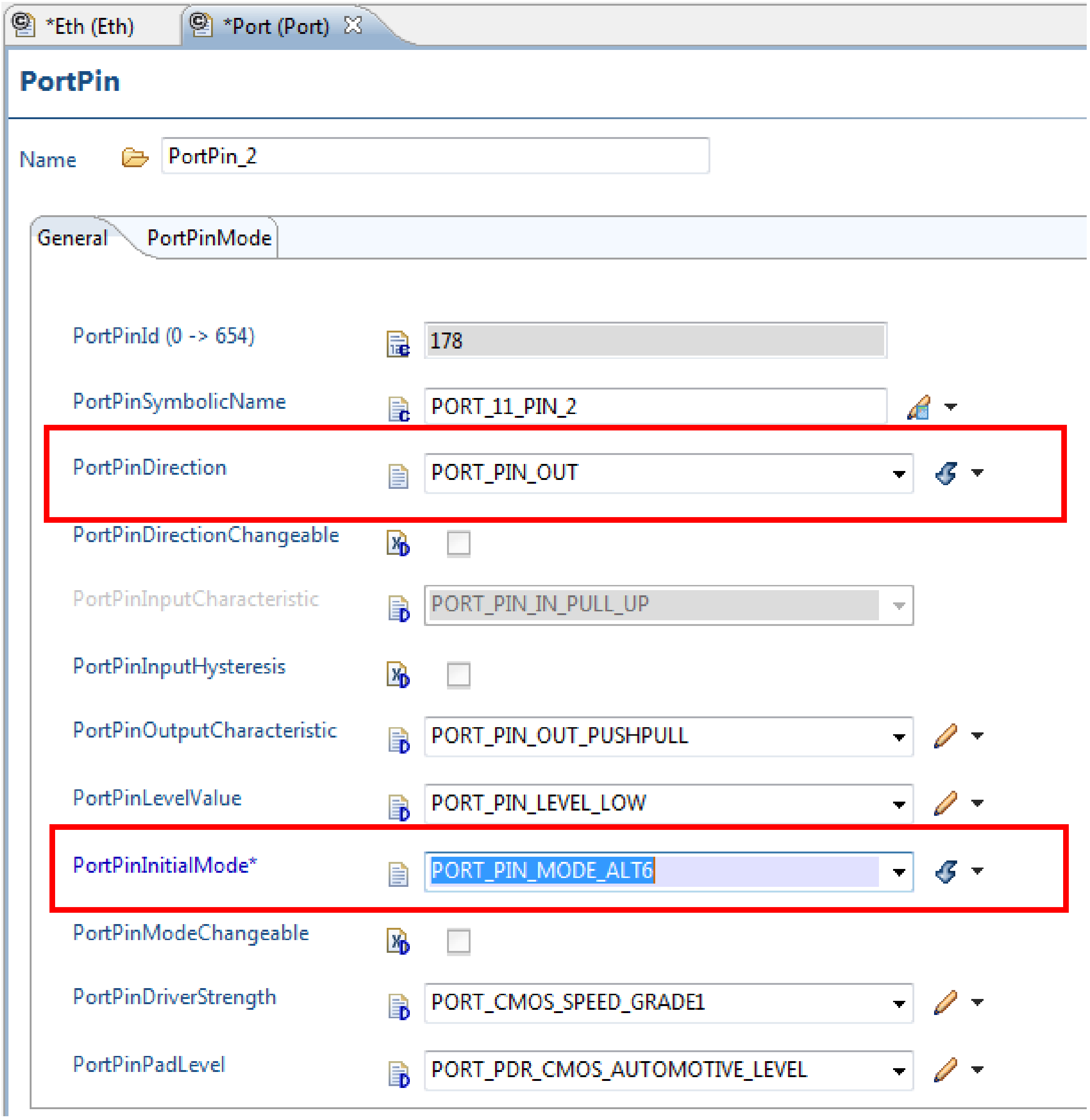
EthCrlEnableMii：Enable/Disable APIs to access PHY;

EthCtrlEnableRxInterrupt/EthCtrlEnableTxInterrupt:使能接收、发送中断函数；

EthCtrlRxBufLenByte/EthCtrlTxBufLenByte/EthRxBufTotal/EthTxBufTotal：定义接收、发送数据的大小；

和 ETH 相关的模式的输出 IO 口要配置成输出状态同时模式也要选对；

例如：当前选择的模式是 RMII，P11.6、P11.3、P11.2 是于 RMII 相关的输出 IO 口，所以 P11.6、P11.3、P11.2 需要在 Port 里面配置成如下模式：



PortPin

Name PortPin_2

PortPinId (0 -> 654) 178

PortPinSymbolicName PORT_11_PIN_2

PortPinDirection PORT_PIN_OUT

PortPinDirectionChangeable □

PortPinInputCharacteristic PORT_PIN_IN_PULL_UP

PortPinInputHysteresis □

PortPinOutputCharacteristic PORT_PIN_OUT_PUSHPULL

PortPinLevelValue PORT_PIN_LEVEL_LOW

PortPinInitialMode* PORT_PIN_MODE_ALT6

PortPinModeChangeable □

PortPinDriverStrength PORT_CMOS_SPEED_GRADE1

PortPinPadLevel PORT_PDR_CMOS_AUTOMOTIVE_LEVEL

4. 程序代码：

```
     volatile uint8 loopvar = 0;
     volatile uint8 EN1Val, EN1SelVal, PortPC12;
     EN1Val = (uint8)SCU_EXTCON.B.EN1;
     EN1SelVal = (uint8)SCU_EXTCON.B.SEL1;
     PortPC12 = (uint8)P11_IOCR12.B.PC12;

     Mcal_ResetSafetyENDINIT_Timed(150000U);
     P11_IOCR12.B.PC12 = 0x16U;
     SCU_EXTCON.B.SEL1 = 5U;
     SCU_EXTCON.B.EN1 = 1U;
     Mcal_SetSafetyENDINIT_Timed();
```

这面这段代码是使能P11.12（就是ETH的ETHREFCLK的功能引脚）输出25MHz的时钟信号，按照工作方式这个引脚应该是检测外部PHY的时钟输出信号，但是这里软件强制输出一个25MHz时钟，

```
     Eth_17_EthMac_ControllerInit(0, 0);
     Eth_17_EthMac_SetControllerMode(0,ETH_MODE_ACTIVE);
     Eth_17_EthMac_GetPhysAddr((uint8)0,&Temp_Macaddress[0]);
```

使能Eth模块的Mac控制器；

```
     Mcal_ResetSafetyENDINIT_Timed(150000U);
     SCU_EXTCON.B.EN1 = EN1Val;
     SCU_EXTCON.B.SEL1 = EN1SelVal;
     P11_IOCR12.B.PC12 = PortPC12;
     Mcal_SetSafetyENDINIT_Timed();
```

P11.12 引脚重新配置成 MCAL 配置的初始化状态；之后使用如下函数发送数据；

```
/******************************************************************************
**
**    Transmit ETH Frame of User Defined Length
**
*******************************************************************************
*/
void Tx_EthFrame (void)
{
  char szString[80];
  uint16 TmpTxlength,i;
  uint8 *TmpEthTxBuffPtr;

  TxFrameCount = 0;



  TmpTxlength = 5;

  Global_TempUSage1 =
Eth_17_EthMac_ProvideTxBuffer(0,&TmpBuffIdx,&TmpEthTxBuffPtr,&TmpTxlength);

  for (i=0; i < 5 ; i++)
  {
     *TmpEthTxBuffPtr = i+1;
     TmpEthTxBuffPtr+=1;
  }

  EthDemoRetVal =
Eth_17_EthMac_Transmit(0,TmpBuffIdx,ETH_FRAME_TYPE_1,1,TmpTxlength,&MacDestArp
address[0]);
  Timer_DelayMs(300);
  #if (!((ETH_17_ENA_TX_INT == STD_ON) && (ETH_17_ENA_RX_INT == STD_ON)))
  do
  {
    Eth_17_EthMac_TxConfirmation(0);
  }
  #endif
  while(!TxFrameCount);
  TxFrameCount = 0;
}
```