```
// This function initializes ePWM4
void FlaxInitEPwm4(Uint16 Period)
{
#define EPWM4_CMPA_INIT 0
#define EPWM4_CMPB_INIT 0

    EALLOW;
    // Time-Base Submodule Register
    EPwm4Regs.TBPRD = Period;                           // Set Timer Period
    EPwm4Regs.TBPHS.all = 0;                            // Set Timer Phase
    EPwm4Regs.TBCTR = 0x0000;                           // Clear counter

    EPwm4Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;          // Count up
    EPwm4Regs.TBCTL.bit.PHSEN = TB_DISABLE;             // Disable phase loading
    EPwm4Regs.TBCTL.bit.PRDLD = TB_SHADOW;              // Load Period from Shadows
    EPwm4Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;          // Synchronization Out Select
    EPwm4Regs.TBCTL.bit.HSPCLKDIV = TB_HSCLK_DIV1;      // PWM Clock = CLK /
    (HSPCLKDIV*CLKDIV)
    EPwm4Regs.TBCTL.bit.CLKDIV = TB_CLK_DIV1;           // PWM Freq = PWMClock / Period
    EPwm4Regs.TBCTL.bit.FREE_SOFT = FREESOFT;           // Set Emulation mode to free run

    // Counter Compare Submodule Register
    EPwm4Regs.CMPA.half.CMPA = 0;                       // Set compare A value
    EPwm4Regs.CMPB = 0;                                 // Set compare B value

    EPwm4Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;         // Set CMPA Immediate or Shadow
    EPwm4Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;         // Set CMPB Immediate or Shadow
    EPwm4Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;       // Set CMPA Shadow Mode
    EPwm4Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;       // Set CMPB Shadow Mode

    // Set Action Qualifier Submodule Register
    EPwm4Regs.AQCTLA.bit.ZRO = AQ_SET;                  // Set PWM1A on Zero
    EPwm4Regs.AQCTLA.bit.PRD = AQ_NO_ACTION;            // NA on PRD
    EPwm4Regs.AQCTLA.bit.CAU = AQ_CLEAR;                // Clear on CMPA - up count
    EPwm4Regs.AQCTLA.bit.CAD = AQ_NO_ACTION;            // NA on CMPA - down count
    EPwm4Regs.AQCTLA.bit.CBU = AQ_NO_ACTION;            // NA on CMPB - up count
    EPwm4Regs.AQCTLA.bit.CBD = AQ_NO_ACTION;            // NA on CMPB - down count

    EPwm4Regs.AQCTLB.bit.ZRO = AQ_SET;                  // Set PWM1B on Zero
    EPwm4Regs.AQCTLB.bit.PRD = AQ_NO_ACTION;            // NA on PRD
    EPwm4Regs.AQCTLB.bit.CAU = AQ_NO_ACTION;            // NA on CMPA - up count
    EPwm4Regs.AQCTLB.bit.CAD = AQ_NO_ACTION;            // NA on CMPA - down count
    EPwm4Regs.AQCTLB.bit.CBU = AQ_CLEAR;                // Clr on CMPB - up count
    EPwm4Regs.AQCTLB.bit.CBD = AQ_NO_ACTION;            // NA on CMPB - down count

    EPwm4Regs.AQSFRC.bit.RLDCSF = SF_LOAD_ON_ZERO;      // SF Action-Qualifier Load on Zero
    EPwm4Regs.AQSFRC.bit.ACTSFA = SF_ACTION_DISABLE;    // SF Action-Qualifier A Disable
    EPwm4Regs.AQSFRC.bit.ACTSFB = SF_ACTION_DISABLE;    // SF Action-Qualifier B Disable

    EPwm4Regs.AQCSFRC.bit.CSFA = SFC_FORCE_DISABLE;     // SFC Action-Qualifier A Disable
    EPwm4Regs.AQCSFRC.bit.CSFB = SFC_FORCE_DISABLE;     // SFC Action-Qualifier B Disable

    // Set Dead-Band Submodule Register
    EPwm4Regs.DBCTL.bit.OUT_MODE = DB_DISABLE;          // Dead-Band Out Mode
    EPwm4Regs.DBCTL.bit.POLSEL = DB_ACTV_HI;            // Dead-Band Polarity Select
    EPwm4Regs.DBCTL.bit.IN_MODE = DBA_ALL;              // Dead-Band Input Select
    EPwm4Regs.DBRED = 0;                                // Dead-Band Rising Edge Delay
    EPwm4Regs.DBFED = 0;                                // Dead-Band Falling Edge Delay

    // Set Trip Zone Submodule Register
    EPwm4Regs.TZSEL.all = 0;                            // Trip Zone Select OSHT1..6 & CBC1..6
    EPwm4Regs.TZCTL.bit.TZA = TZ_NO_CHANGE;             // Trip Zone Control A channel
    EPwm4Regs.TZCTL.bit.TZB = TZ_NO_CHANGE;             // Trip Zone Control B channel
    EPwm4Regs.TZEINT.bit.CBC = TZ_DISABLE;              // Trip Zone Cycle by Cycle Int Enable
    EPwm4Regs.TZEINT.bit.OST = TZ_DISABLE;              // Trip Zone One-Shot Int Enable

    // Set Event Trigger Register
    EPwm4Regs.ETSEL.bit.INTSEL = ET_CTR_ZERO;           // PWM Int Event on CTR == ZERO
    EPwm4Regs.ETSEL.bit.INTEN = X_DISABLE;              // PWM Int Event Disable
    EPwm4Regs.ETSEL.bit.SOCASEL = ET_CTR_ZERO;          // PWM SOCA Event on CTR == ZERO
    EPwm4Regs.ETSEL.bit.SOCAEN = X_DISABLE;             // PWM SOCA Event Disable
    EPwm4Regs.ETSEL.bit.SOCBSEL = ET_CTR_ZERO;          // PWM SOCB Event on CTR == ZERO
    EPwm4Regs.ETSEL.bit.SOCBEN = X_DISABLE;             // PWM SOCB Event Disable

    EPwm4Regs.ETPS.bit.INTPRD = ET_1ST;             // Interrupt Period Select
    EPwm4Regs.ETPS.bit.INTCNT = ET_1ST;             // Interrupt Count Select
    EPwm4Regs.ETPS.bit.SOCAPRD = ET_DISABLE;            // SOCA Period Select
    EPwm4Regs.ETPS.bit.SOCACNT = ET_DISABLE;            // SOCA Count Select
    EPwm4Regs.ETPS.bit.SOCBPRD = ET_DISABLE;            // SOCB Period Select
    EPwm4Regs.ETPS.bit.SOCBCNT = ET_DISABLE;            // SOCB Count Select

    // Set HRPWM Register
    EPwm4Regs.HRCNFG.bit.EDGMODE = HR_Disable;          // HR is Disable
    EPwm4Regs.HRCNFG.bit.CTLMODE = HR_CMP;              // HR on Compare
    EPwm4Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO;          // Shadows Load on Zero

    EDIS;
}
/****************************************************************************/
// This function initializes eCAP1
```