ISP

Введение в новую экосистему взаимодействия программных библиотек Microchip

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Цели класса

По окончании класса Вы сможете:

- **Объяснить необходимость платформы для взаимодействия библиотек**
- **Понять основные концепции организации платформы Microchip нового поколения.**
- **Понять преимущества от использования платформы Microchip нового поколения**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- **Недостатки традиционного подхода**
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- **Введение в Harmony**
- **Заключение**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- **Недостатки традиционного подхода**
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- **Введение в Harmony**
- **Заключение**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

MICROCHIP MASTERs 2013

Традиционный подход

```
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include <xc.h>

#define APP_LED_BLINKING_RATE_1S      0x0004C4B4
#define APP_LED_BLINKING_RATE_500mS   0x0002625A
#define APP_LED_BLINKING_RATE         APP_LED_BLINKING_RATE_1S

#pragma config FPLLODIV = DIV_1, FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FWDTEN = OFF, FCKSM = CSECME, FPBDIV = DIV_1
#pragma config OSCIOFNC = ON, POSCMOD = HS, FSOSCEN = ON, FNOSC = PRIPLL
#pragma config ICESEL = ICS_PGx2

int main ( void )
{
    TRISA = 0;
    LATA = 0;
    T2CON = 0;
    T3CON = 0;
    T2CONbits.ON = 0;
    T2CONbits.TCKPS = 7;
    T2CONbits.TCS = 0;
    T2CONbits.T32 = 1;
    TMR2 = 0;
    PR2 = APP_LED_BLINKING_RATE;
    T2CONbits.ON = 1;

    while(true)
    {
        if ( IFS0bits.T3IF )
        {
            IFS0bits.T3IF = 0;
            LATAbits.LATA0 = ~LATAbits.LATA0;
        }
    }
    return (-1);
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Аппаратно-зависимые регистры

Регистры зависят от микросхемы

- Имена
- Адреса
- Значения
- Назначение

```
/* Timer Peripheral Initialization */
T2CONbits.ON = 0;
T2CONbits.TCKPS = 7;
T2CONbits.TCS = 0;
T2CONbits.T32 = 1;
TMR2 = 0;
PR2 = APP_LED_BLINKING_RATE;
T2CONbits.ON = 1;

/* Loop forever, blinking the LED. */
while(true)
{
    if ( IFS0bits.T3IF )
    {
        IFS0bits.T3IF = 0;
        LATAbits.LATA0 = ~LATAbits.LATA0;
    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Аппаратно-зависимые регистры

- Надо знать что значат регистры
- Необходимо реализовать всю логику управления периферией

```
/* Timer Peripheral Initialization */
T2CONbits.ON = 0;
T2CONbits.TCKPS = 7;
T2CONbits.TCS = 0;
T2CONbits.T32 = 1;
TMR2 = 0;
PR2 = APP_LED_BLINKING_RATE;
T2CONbits.ON = 1;

/* Loop forever, blinking the LED. */
while(true)
{
    if ( IFS0bits.T3IF )
    {
        IFS0bits.T3IF = 0;
        LATAbits.LATA0 = ~LATAbits.LATA0;
    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Аппаратно-зависимые параметры

```
/* Delays for timer */
#define APP_LED_BLINKING_RATE_1S        0x0004C4B4
#define APP_LED_BLINKING_RATE_500mS     0x0002625A
#define APP_LED_BLINKING_RATE           APP_LED_BLINKING_RATE_1S

/* System clock = 80 MHz, Peripheral clock = 80 MHz */
#pragma config FPLLODIV = DIV_1, FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FWDTEN = OFF
#pragma config OSCIOFNC = ON, POSCMOD = HS, FSOSCEN = ON, FNOSC = PRIPLL
#pragma config ICESEL = ICS_PGx2
```

Численные значения параметров

- Зависят от приложения
- Зависят от микросхемы

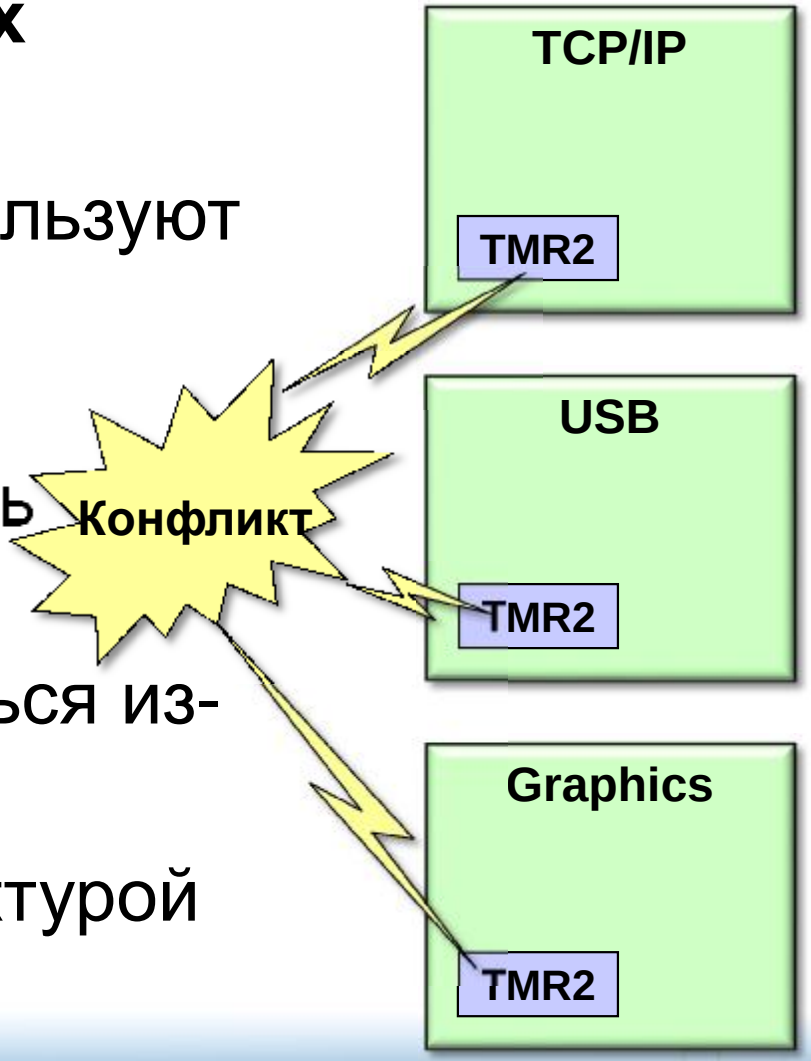
Приводятся только в примерах.
Не могут содержаться в библиотеке.

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Немодульные библиотеки

Конфликты разделяемых ресурсов

- Разные библиотеки используют один ресурс
- Доступ к одним и тем же регистрам может вызвать конфликт
- Может не компилироваться из-за конфликта имен
- Решается сложной структурой из #ifdef'-ов



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

MICROCHIP MASTERs 2013

Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Переносимость

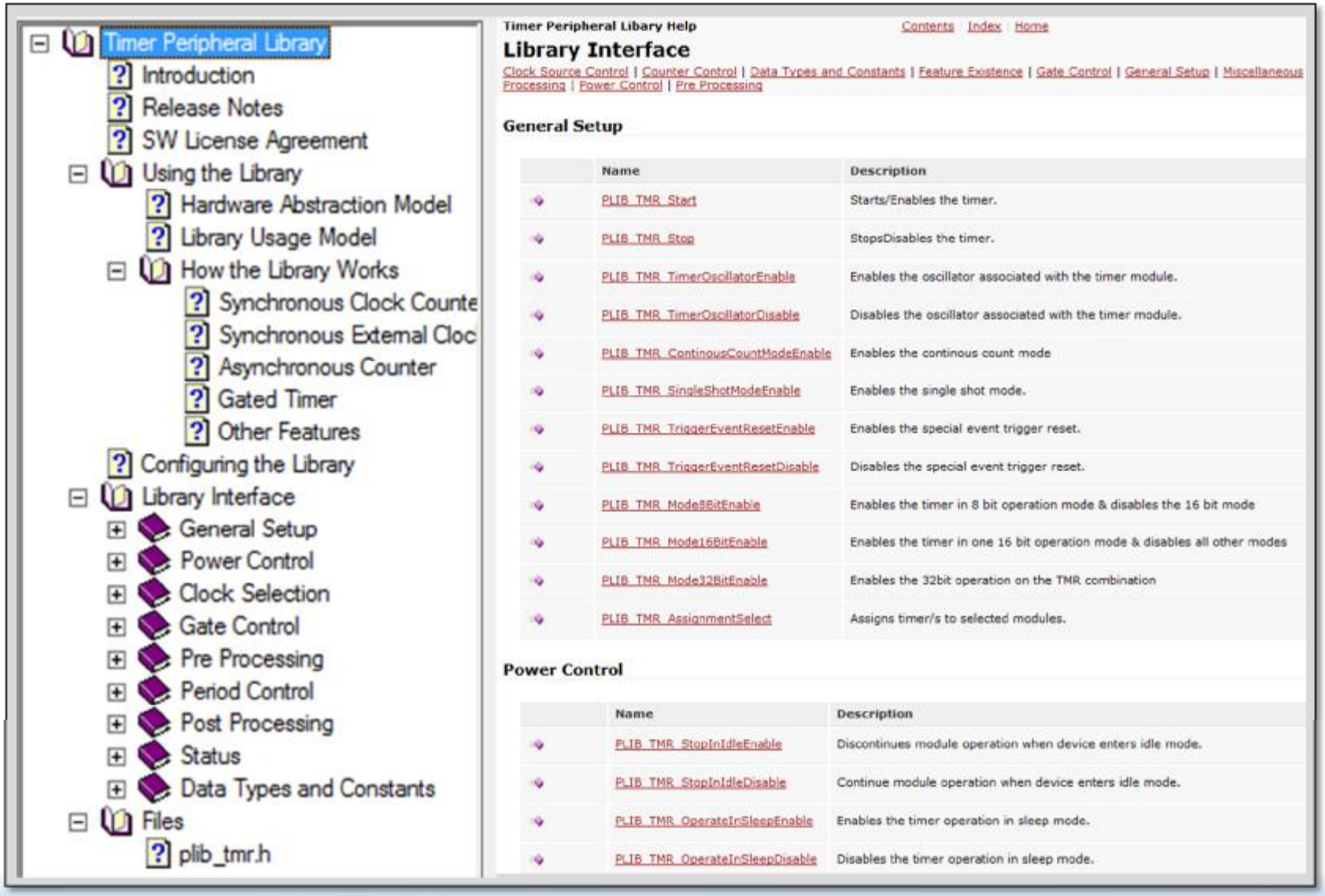
Выбор решений от Microchip

- Широкий диапазон 8-, 16-, & 32-bit микроконтроллеров
- Позволяет оптимизировать цену и параметры

Мы минимизируем стоимость переноса программ между микроконтроллерами Microchip.

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Единый API



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Переносимый код приложения

```
int main ( void )
{
  /* Port Pin Initialization */
  PLIB_PORTS_Write(APP_PORT, APP_PORT_CHANNEL, 0);
  PLIB_PORTS_PinDirectionOutputSet(APP_PORT, APP_P

  /* Timer Peripheral Initialization */
  PLIB_TMR_Stop(APP_TIMER);
  PLIB_TMR_PrescaleSelect(APP_TIMER, APP_TIMER_PRE
  PLIB_TMR_ClockSourceSelect(APP_TIMER, APP_TIMER_CLOCK_SOURC);
  PLIB_TMR_Period32BitSet(APP_TIMER, APP_LED_BLINKING_RATE);
  PLIB_TMR_Mode32BitEnable(APP_TIMER);
  PLIB_TMR_Start(APP_TIMER);

  /* Loop forever, blinking the LED. */
  while(true)
  {
      if( PLIB_INT_SourceFlagGet(APP_INT_CONTROLLER, APP_INT_SOURCE) )
      {
          PLIB_INT_SourceFlagClear(APP_INT_CONTROLLER, APP_INT_SOURCE);
          PLIB_PORTS_PinToggle(APP_PORTS, APP_PORT_CHANNEL, APP_PORT_BIT);
      }
  }
  return(EXIT_VALUE);
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Переносимый код приложения

```
int main ( void )
{
  /* Port Pin Initialization */
  PLIB_PORTS_Write(APP_PORT, APP_PORT_CHANNEL, 0);
  PLIB_PORTS_PinDirectionOutputSet(APP_PORT, APP_P

  /* Timer Peripheral Initialization */
  PLIB_TMR_Stop(APP_TIMER);
  PLIB_TMR_PrescaleSelect(APP_TIMER, APP_TIMER_PRE
  PLIB_TMR_ClockSourceSelect(APP_TIMER, APP_TIMER_CLOCK_SOURCE);
  PLIB_TMR_Period32BitSet(APP_TIMER, APP_LED_BLINKING_RATE);
  PLIB_TMR_Mode32BitEnable(APP_TIMER);
  PLIB_TMR_Start(APP_TIMER);

  /* Loop forever, blinking the LED. */
  while(true)
  {
      if( PLIB_INT_SourceFlagGet(APP_INT_CONTROLLER, APP_INT_SOURCE) )
      {
          PLIB_INT_SourceFlagClear(APP_INT_CONTROLLER, APP_INT_SOURCE);
          PLIB_PORTS_PinToggle(APP_PORTS, APP_PORT_CHANNEL, APP_PORT_BIT);
      }
  }
  return(EXIT_VALUE);
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

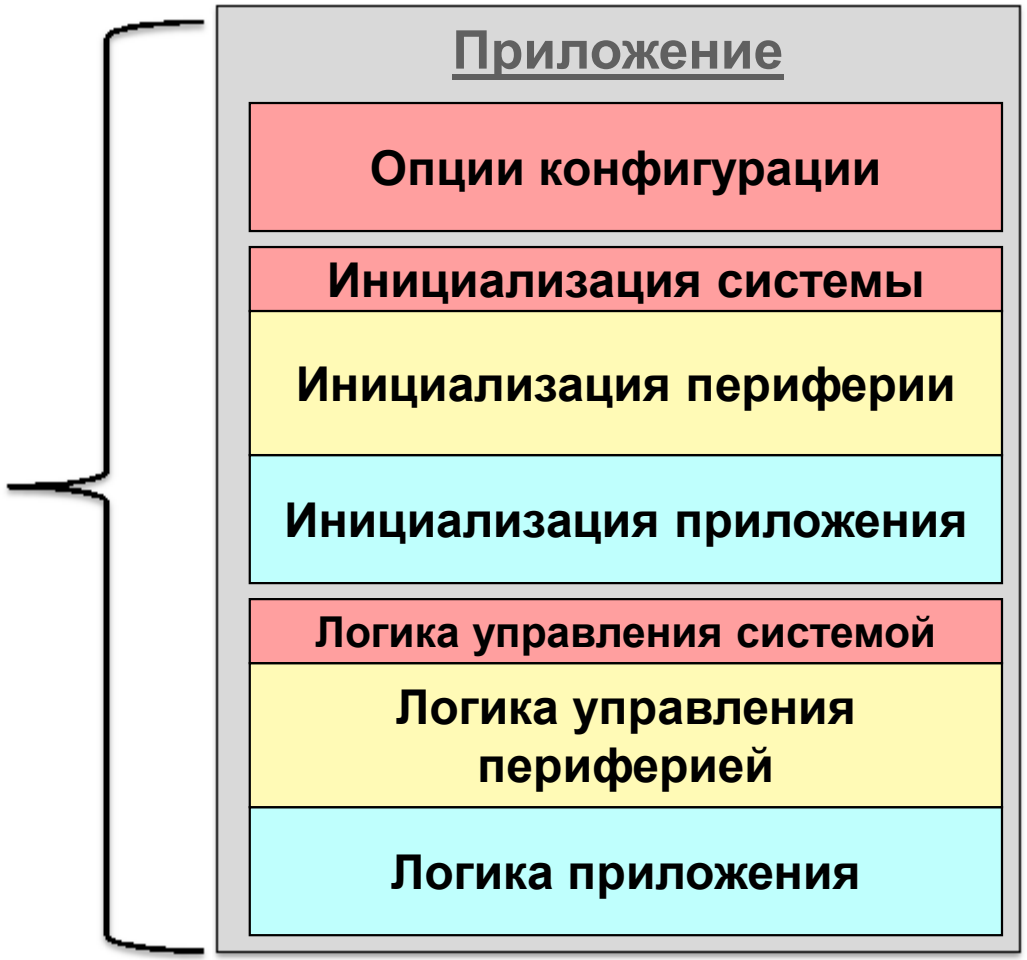
Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

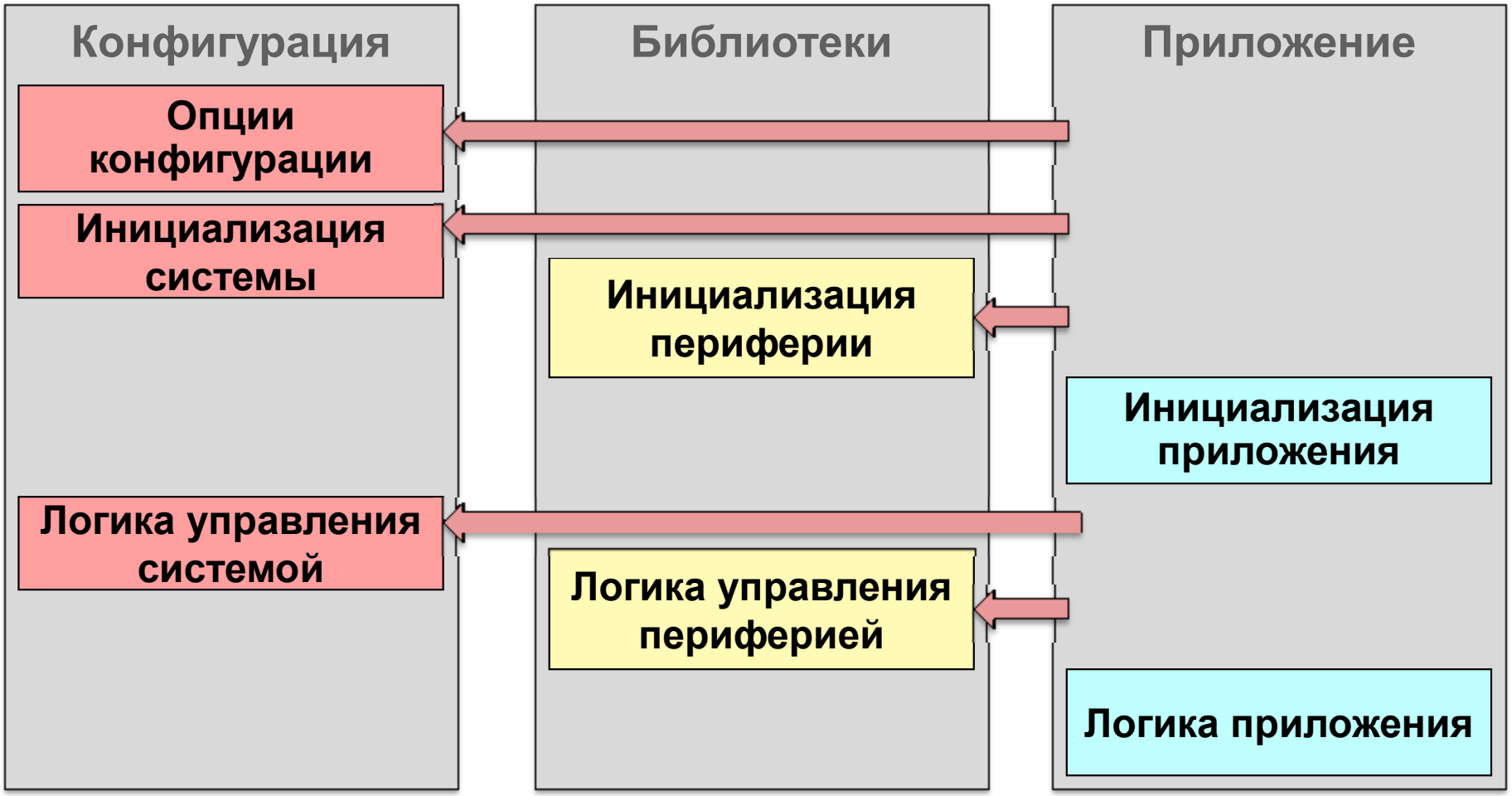
Части приложения

Различные части встраиваемого приложения служат разным целям.



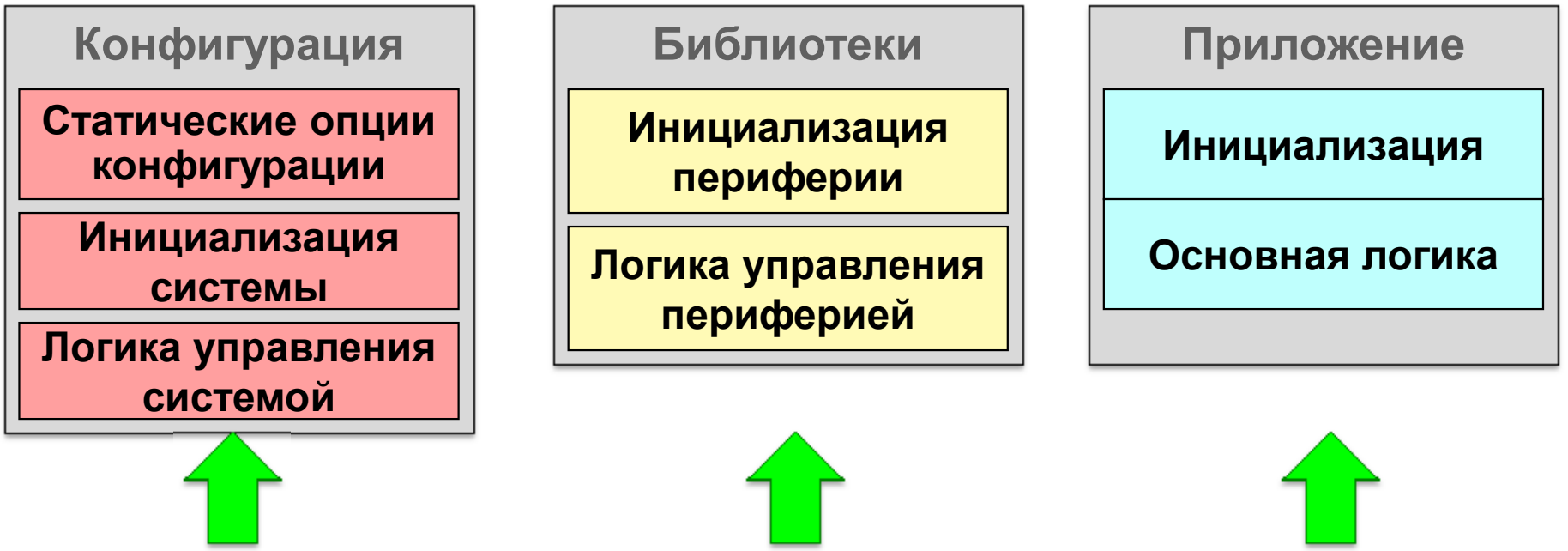
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Части приложения



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Части приложения



Код конфигурации системы

Библиотеки Microchip

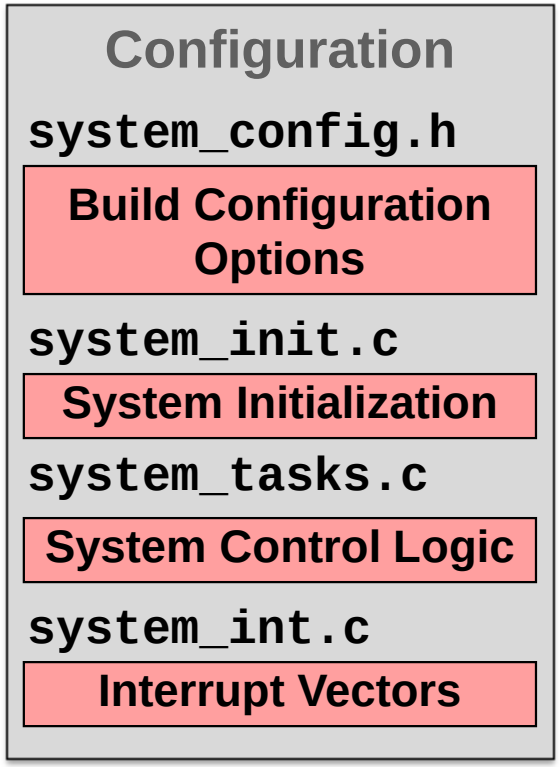
Ваш код приложения

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Конфигурация системы

Файлы конфигурации системы

- **system_config.h**
  - Задает все статические опции
  - Включается всеми библиотеками
- **system_init.c**
  - Биты конфигурации процессора
  - Инициализация всех библиотек и приложения
- **system_tasks.c**
  - Вызывает периодические задачи
  - Обеспечивает машину состояний системы
- **system_int.c**
  - Реализует «обертку» для векторов прерываний
  - Вызывает все обработчики прерываний (ISRs)



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Конфигурация системы

system_config.h

```
#define APP_LED_BLINKING_RATE_1S      0x0004C4B4
#define APP_LED_BLINKING_RATE_500mS   0x0002625A
#define APP_LED_BLINKING_RATE         APP_LED_BLINKING_RATE_1S
```

system_init.c

```
#pragma config FPLLODIV = DIV_1, FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FWDTEN = OFF
#pragma config OSCIOFNC = ON, POSCMOD = HS, FSOSCEN = ON, FNOSC = PRIPLL
#pragma config ICESEL = ICS_PGx2

void SYS_Initialize( void )
{
    /* Вызывает подпрограммы инициализации всех библиотек и приложений */
}
```

```
int main(void)
{
    SYS_Initialize();

    while(true)
    {
        SYS_Tasks();
    }

    return(EXIT_VALUE);
}
```

system_tasks.c

```
void SYS_Tasks( void )
{
    /* Вызывает «задачи» поллинга всех библиотек */
}
```

system_int.c

```
void __ISR ( _TIMER_3_VECTOR ) _ISR_TMR_3_stub ( void )
{
    /* Вызывает обработчики прерывания от таймера
всех      используемых библиотек */
}
```

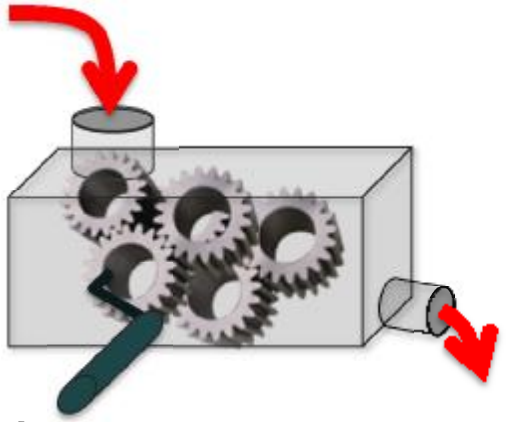
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

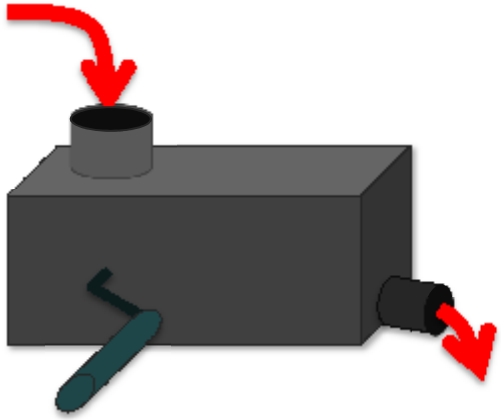
Модульность



- **Доступ через функции интерфейса**
  - Ввод, Обработка, Вывод
  - Может делать «побочную работу»

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Модульность



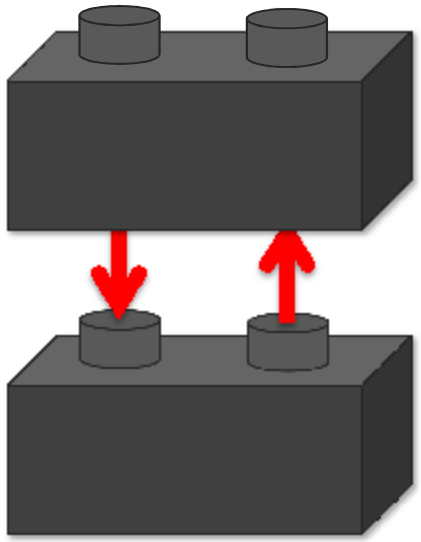
- **Доступ через функции интерфейса**
  - Ввод, Обработка, Вывод
  - Может делать «побочную работу»
  - Рассматривается как «черный ящик»
- **Модуль**
  - Одна или несколько тесно связанных функций
  - Разделяют общие данные и ресурсы
  - Обрабатывают состояние общих ресурсов

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Модульность

Модули

- **Функции интерфейса**
  - Нацелены на выполнение конкретной функциональности
  - Минимально достаточный набор
- **Почти не зависят друг то друга**
  (Взаимодействуют через межмодульный интерфейс)
- **Разделяют решение разных задач**



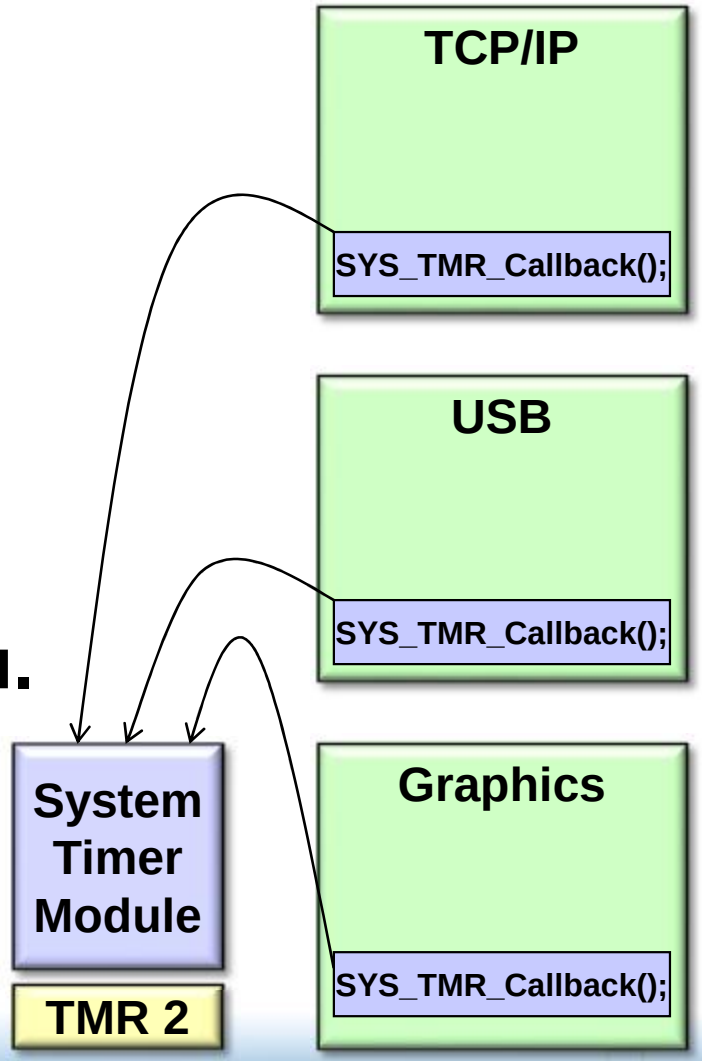
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Модульность

Каждый модуль…

- …владеет и управляет своими ресурсами,
- …обеспечивает простой абстрактный интерфейс,
- …для использования других ресурсов вызывает другие модули.

Это исключает конфликты !



© 2013 Microchip Technology Incorporated. All Rights Reserved.

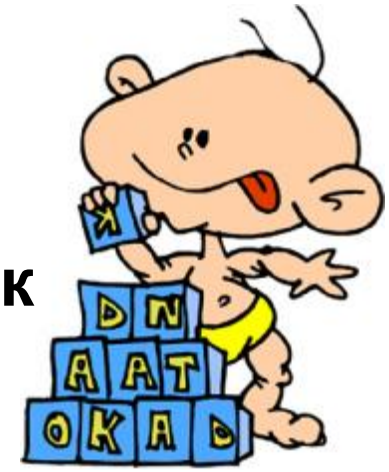
MICROCHIP
MASTERs 2013

Модульность

```
SYS_TMR_HANDLE SYS_TMR_CallbackPeriodic (unsigned int period, SYS_TMR_CALLBACK callback)
{
    /* Variable to hold the Queue element index */
    int8_t qElementIndex = SYS_TMR_HANDLE_INVALID;

    /* Check to see if the request will fit in the queue */
    if (sIndex < SYS_TMR_MAX_PERIODIC_EVENTS)
    {
        /* Assign the event parameters */
        sCallbackObject[sIndex].period   = ( period / (sSysTmrObject.alarmPeriod) );
        sCallbackObject[sIndex].elapsed  = 0;
        sCallbackObject[sIndex].count    = 0;
        sCallbackObject[sIndex].status   = SYS_TMR_CALLBACK_ACTIVE;
        sCallbackObject[sIndex].type     = SYS_TMR_CALLBACK_PERIODIC;
        sCallbackObject[sIndex].callback = callback;

        /* Register the event */
        qElementIndex = QUEUE_Push( &sSysTmrObject.eventQ, &sCallbackObject[sIndex] );
        sSysTMRSingleEventStatus = true;
        sIndex++;
    }
    else
    {
        qElementIndex = SYS_TMR_HANDLE_INVALID;
    }

    /* Return the callback handle */
    return ( qElementIndex );
}
```

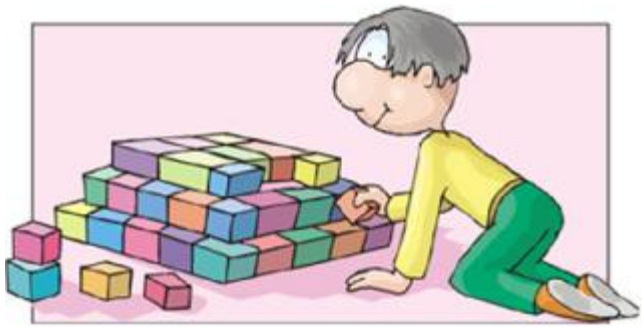
Модульность

Модульные библиотеки являются «строительными блоками» приложения



Ускорение времени выхода на рынок

- Ускоряет разработку
- Облегчает добавление новых функций



- Снижает затраты на поддержку
- Позволяет быстро отвечать на изменения рынка

© 2013 Microchip Technology Incorporated. All Rights Reserved.

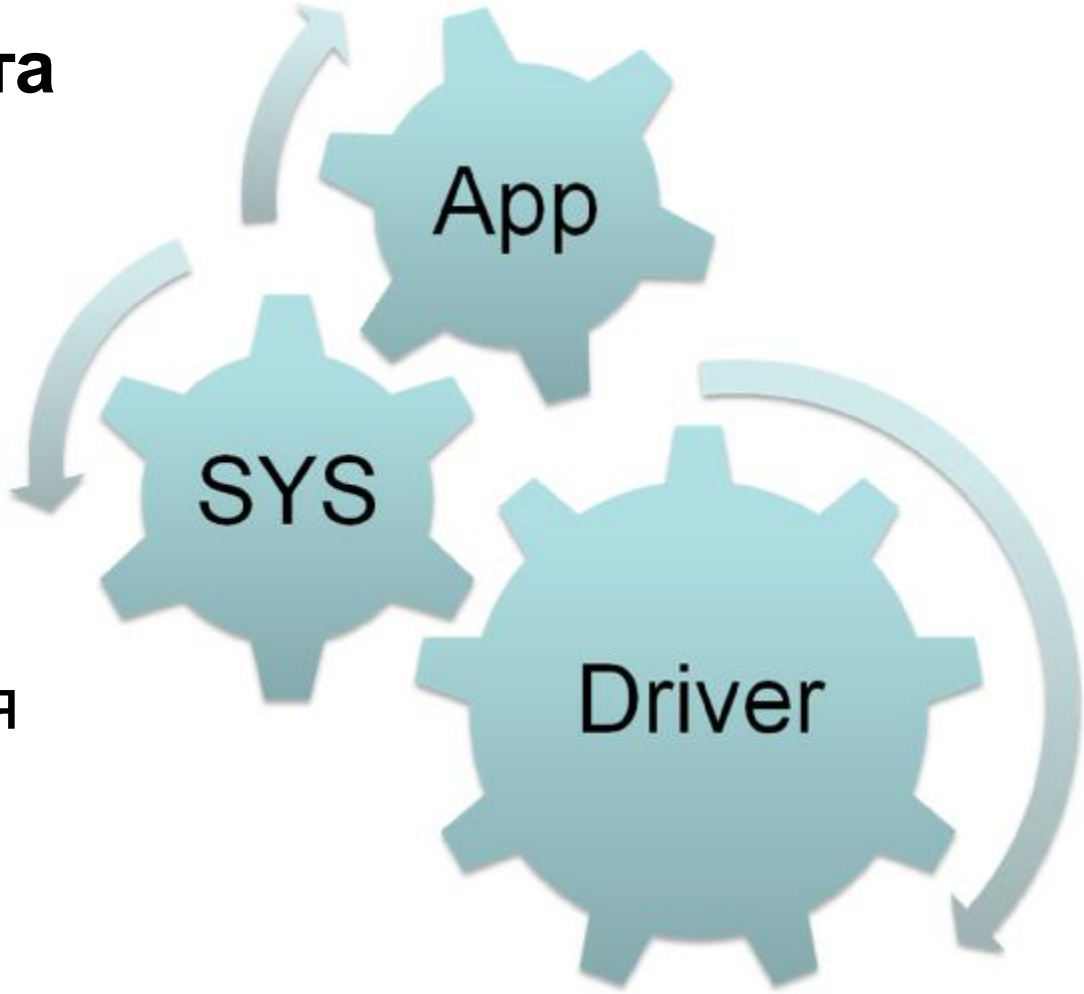
Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимость

- **Совместная работа модулей**
- **Базируется на машинах состояний**
- **Системные функции:**
  - Инициализация модулей
  - Вызов “задач” модулей



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

```
main()
{
        /* Вызов функции инициализации State Machine 1 */
        /* Вызов функции инициализации State Machine 2 */
        /* Вызов функции инициализации State Machine 3 */

            while(true);
            {
                        /* Вызов функции обработки задачи
                           State Machine 1 */
                        /* Вызов функции обработки задачи
                           State Machine 2 Task */
                        /* Вызов функции обработки задачи
                           State Machine 3 Task */

            }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

```
while(true);
{
    if ( /* Условие Task 1 */ )
    {
        /* Выполнить Task 1 */
    }

    if ( /* Условие Task 2 */ )
    {
        /* Выполнить Task 2 */
    }

    if ( /* Условие Task 3 */ )
    {
        /* Выполнить Task 3 */
    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

```
while(true);
{
    if ( /* Условие Task 1 */ )
    {
        /* Выполнить Task 1 */
    }

    if ( /* Условие Task 2 */ )
    {
        /* Выполнить Task 2 */
    }

    if ( /* Условие Task 3 */ )
    {
        /* Выполнить Task 3 */
    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

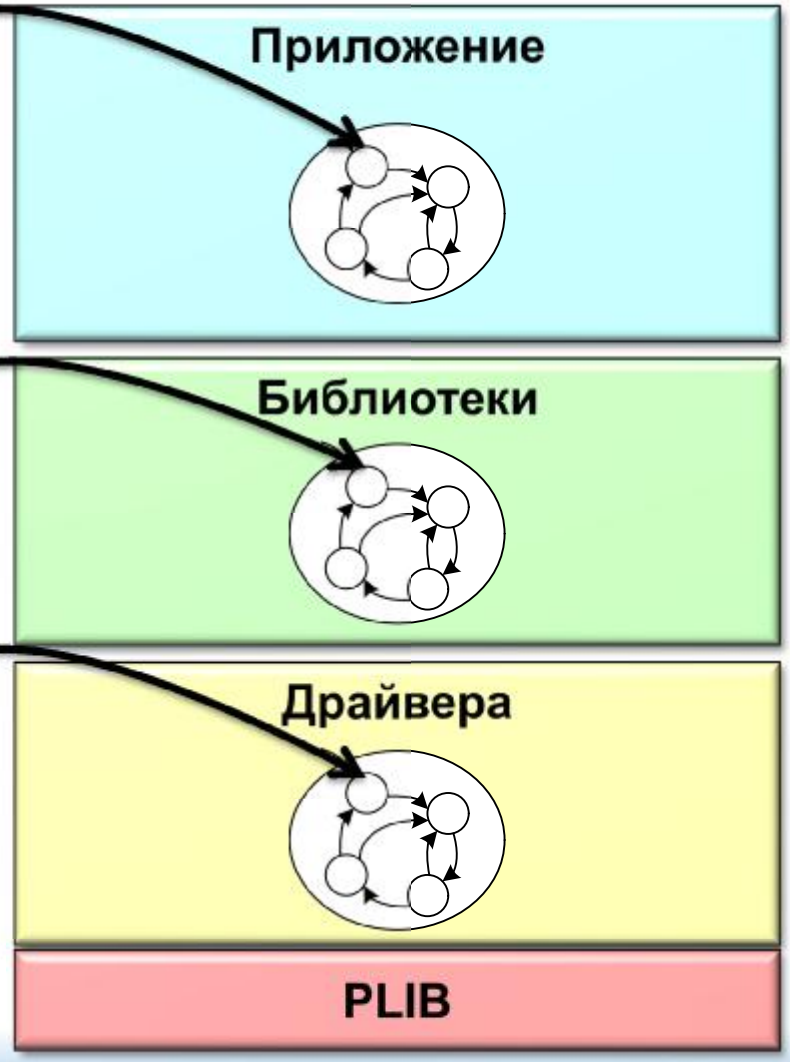
```
while(true);
{
    if ( /* Условие Task 1 */ )
    {
        /* Выполнить Task 1 */
    }

    if ( /* Условие Task 2*/ )
    {
        /* Выполнить Task 2 */
    }

    if ( /* Условие Task 3 */ )
    {
        /* Выполнить Task 3 */
    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

Абсолютно независимы

Нет связей, зависящих от состояния или временных соотношений других модулей



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Инициализация системы и выполнение задач

```
int main(void)
{
    SYS_Initialize(NULL);

    while(true)
    {
        SYS_Tasks();
    }

    retur
}
```

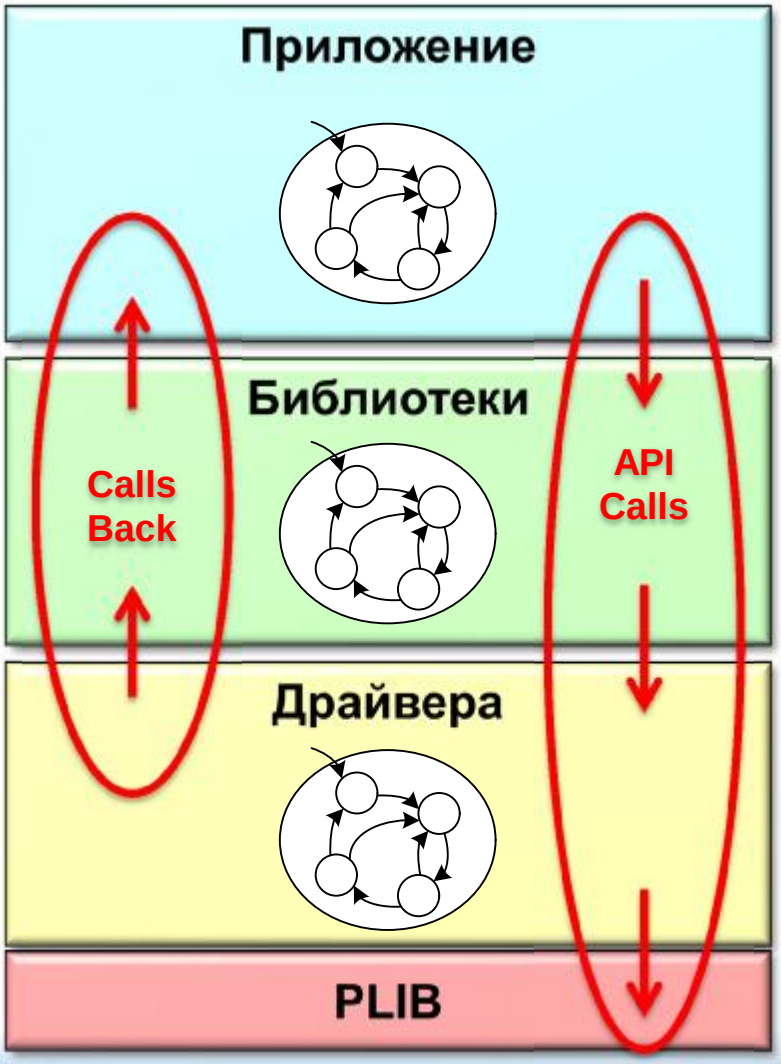
```
void SYS_Initialize( void* data )
{
    SYS_CLK_Initialize( &clkInit );
    BSP_Initialize();
    modules.usart  = DRV_USART_Initialize( APP_DRV_USART_INDEX, &usartInit );
    modules.tmr    = DRV_TMR_Initialize(   APP_DRV_TMR_INDEX,   &tmrInit );
    modules.sysTmr = SYS_TMR_Initialize(   APP_SYS_TMR_INDEX,   &sysTmrInit );

    APP_Initialize();
}
```

```
void SYS_Tasks ( void )
{
    DRV_USART_TasksTX( modules.usart );
    DRV_TMR_Tasks(     modules.tmr );
    SYS_TMR_Tasks(     modules.sysTmr );

    APP_Tasks();
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

Модули взаимодействуют друг с другом только через функции интерфейса



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Совместимые машины состояний

```
void APP_Tasks ( void )
{
    DRV_USART_CLIENT_STATUS usartStatus;

    switch ( app.state )
    {
        case APP_STATE_INIT:
            app.usart = DRV_USART_Open( APP_DRV_USART_INDEX, DRV_IO_INTENT_READWRITE );
            if ( app.usart != DRV_USART_HANDLE_INVALID )
            {
                app.state = APP_STATE_WAIT_FOR_INIT;
            }
            break;

        case APP_STATE_WAIT_FOR_INIT:
            usartStatus = DRV_USART_ClientStatus ( app.usart );
            if ( usartStatus == DRV_USART_CLIENT_STATUS_READY )
            {
                app.state = APP_STATE_USART_MESSAGE_BEGIN;
            }
            break;

        //...

    }
}
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- Недостатки традиционного подхода
- **Основные концепции нового подхода**
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

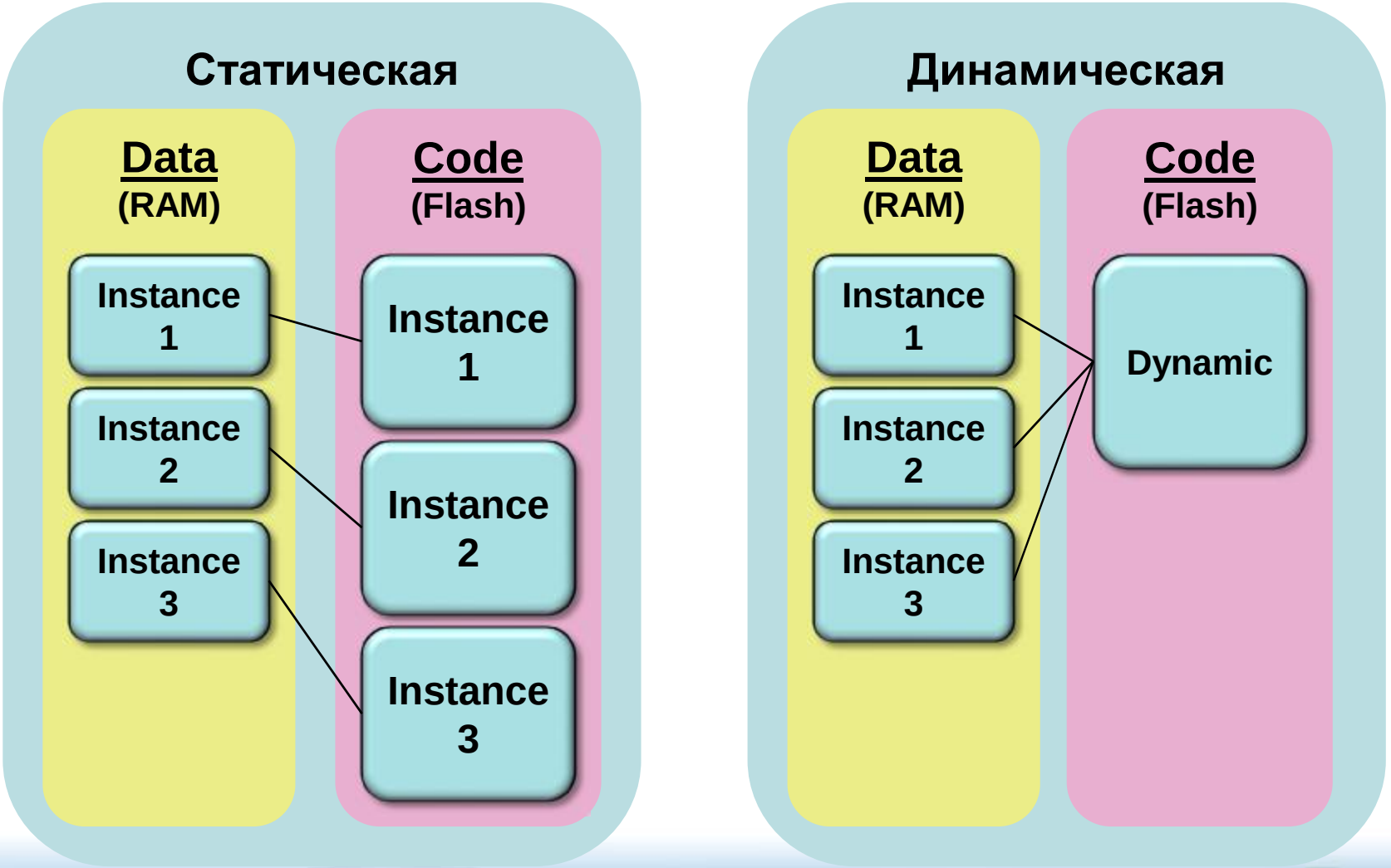
Гибкость

Возможные конфигурации

- **Реализации**
  - Статическая/Динамическая
  - Одиночный/Множественный клиент
- **Без OS**
  - Поллинг
  - По прерываниям
- **На базе RTOS**

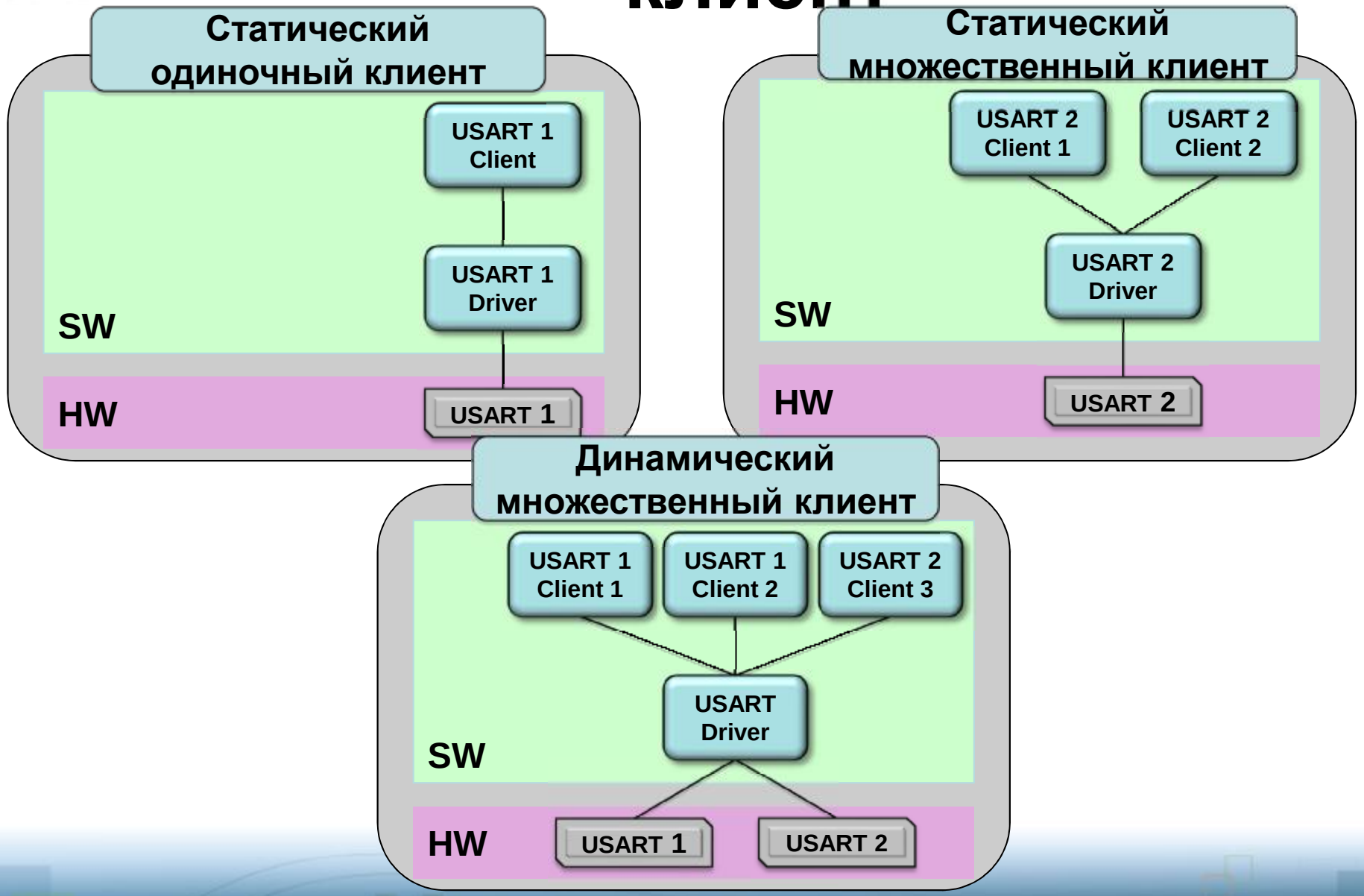
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Статическая и динамическая реализации



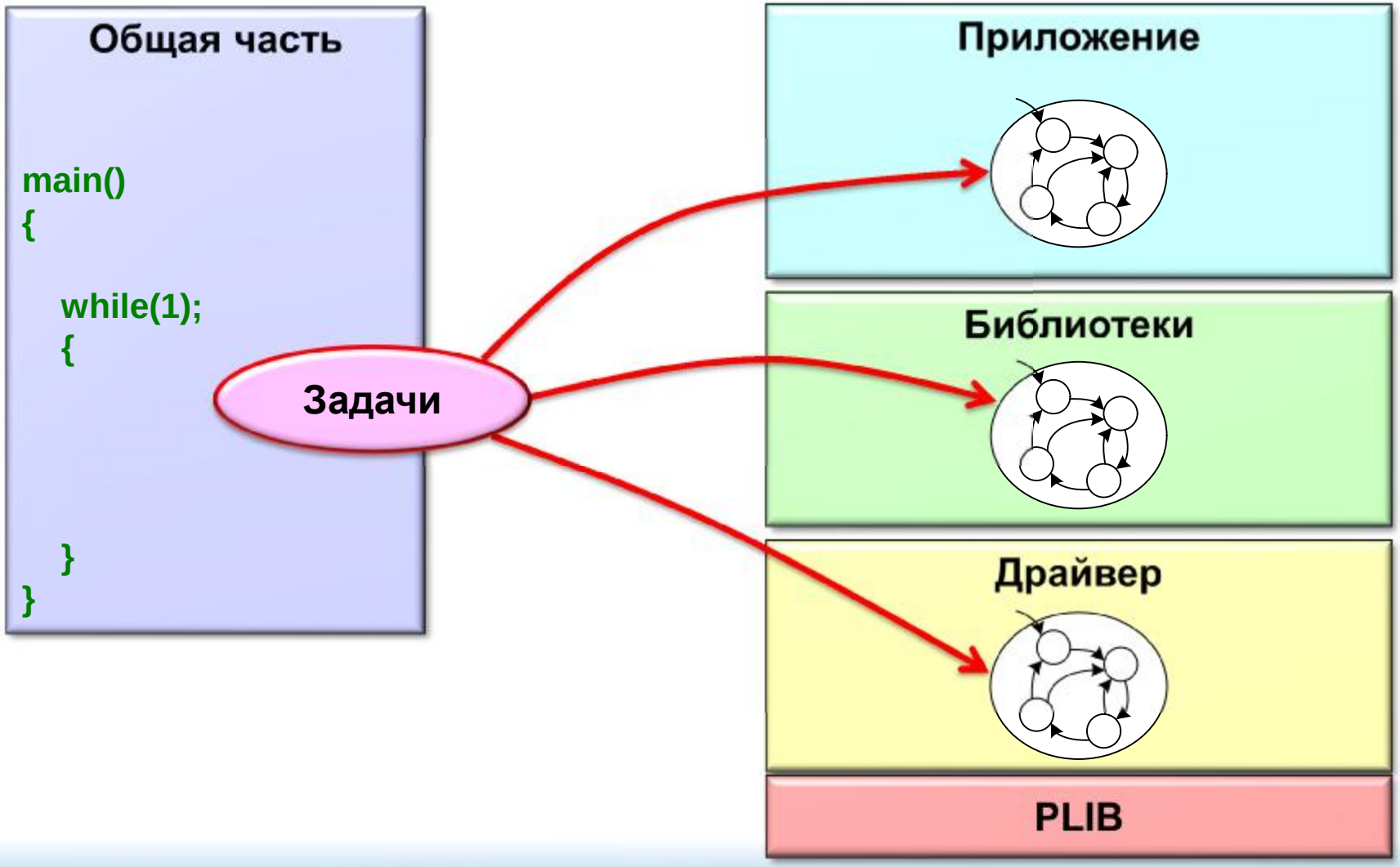
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Одиночный/Множественный клиент



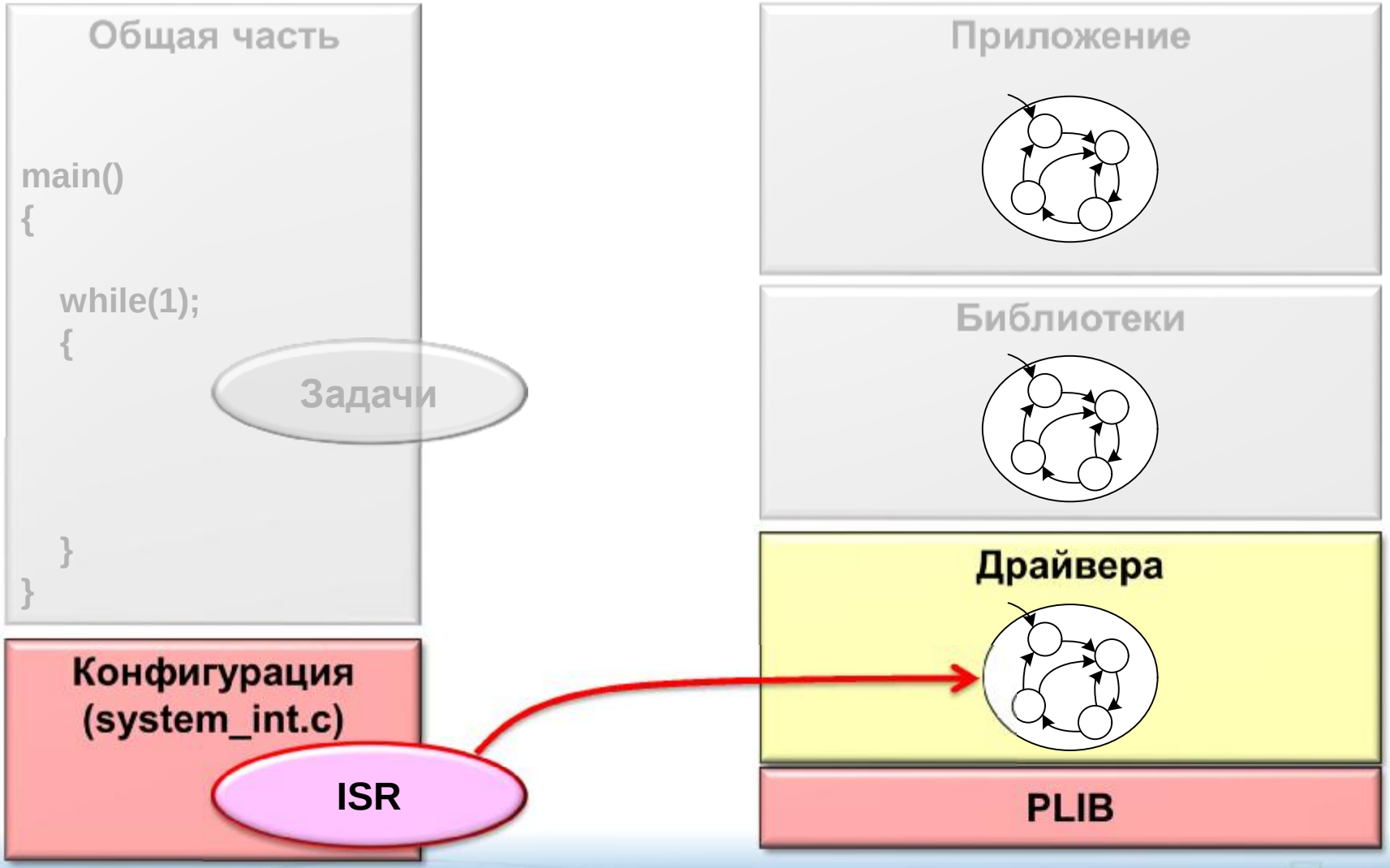
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Поллинг



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Работа по прерываниям



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Работа по прерываниям

```
void __ISR ( _TIMER_3_VECTOR ) _InterruptHandler_TMR_3_stub( void )
{
    /* Call the timer driver's "Tasks" routine */
    DRV_TMR_Tasks( appDrvObject.tmrDrvObject );
}
```

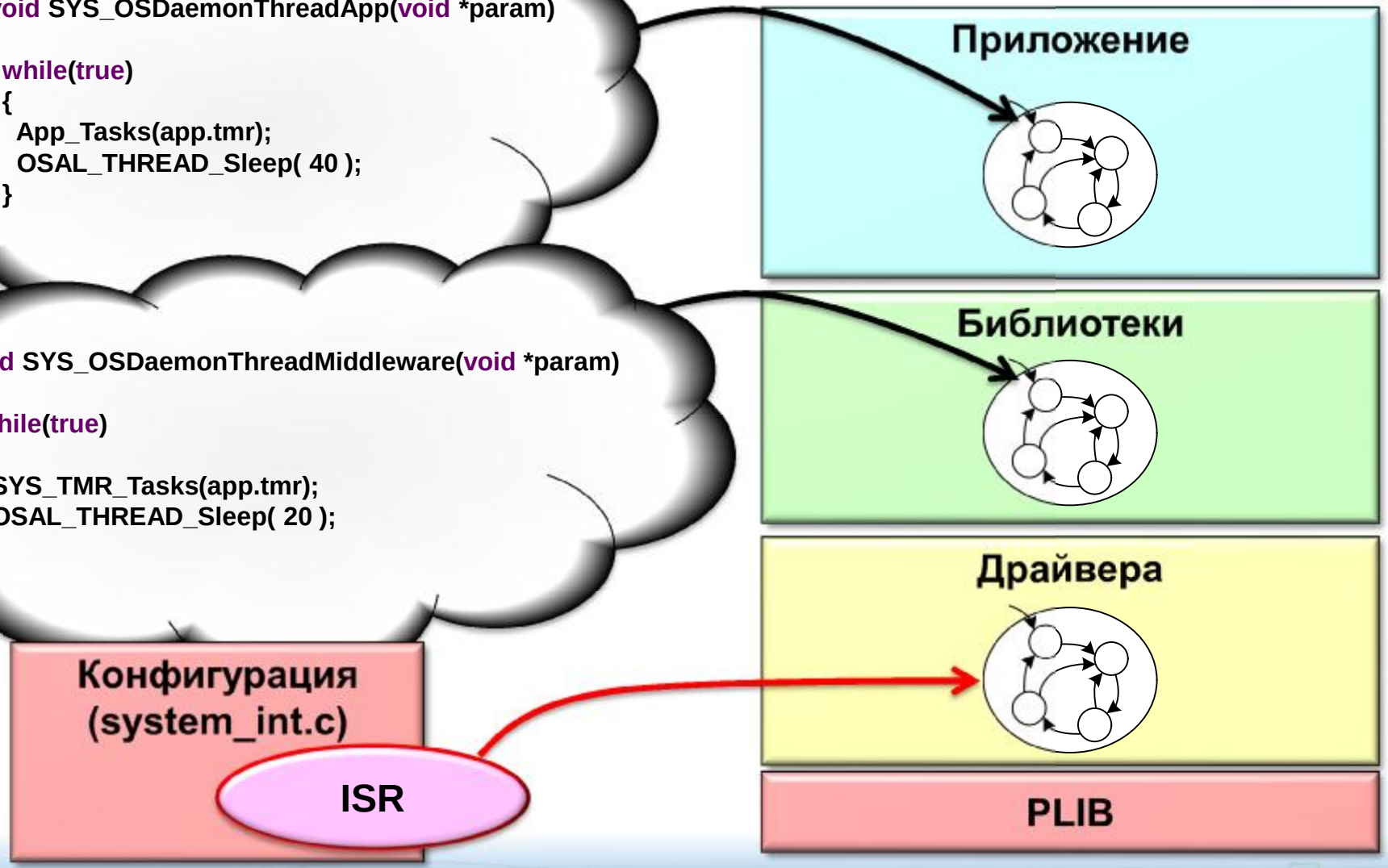
```
void DRV_TMR_Tasks( SYS_MODULE_OBJ object )
{
    DRV_TMR_OBJ *dObj = (DRV_TMR_OBJ *) object;

    if ( true == SYS_INT_SourceStatusGet(dObj->interruptSource) )
    {
        _DRV_TMR_PeriodSet(dObj->tmrId, dObj->timerPeriod);
        dObj->elapseStatus = true;

        /* Call the client back if the alarm is active */
        if ( dObj->alarmInUse == true )
        {
            dObj->alarmCount = dObj->alarmCount + 1;
            if( dObj->alarmCallback != NULL )
            {
                dObj->alarmCallback();
            }
        }

        /* Clear Timer Interrupt/Status Flag */
        SYS_INT_SourceClear(dObj->interruptSource);
```

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Взаимодействие с RTOS

```
void SYS_OSDaemonThreadApp(void *param)
{
  while(true)
  {
    App_Tasks(app.tmr);
    OSAL_THREAD_Sleep( 40 );
  }
}
```

```
void SYS_OSDaemonThreadMiddleware(void *param)
{
  while(true)
  {
    SYS_TMR_Tasks(app.tmr);
    OSAL_THREAD_Sleep( 20 );
  }
}
```



© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- Недостатки традиционного подхода
- Основные концепции нового подхода
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- **Введение в Harmony**
- Заключение

© 2013 Microchip Technology Incorporated. All Rights Reserved.

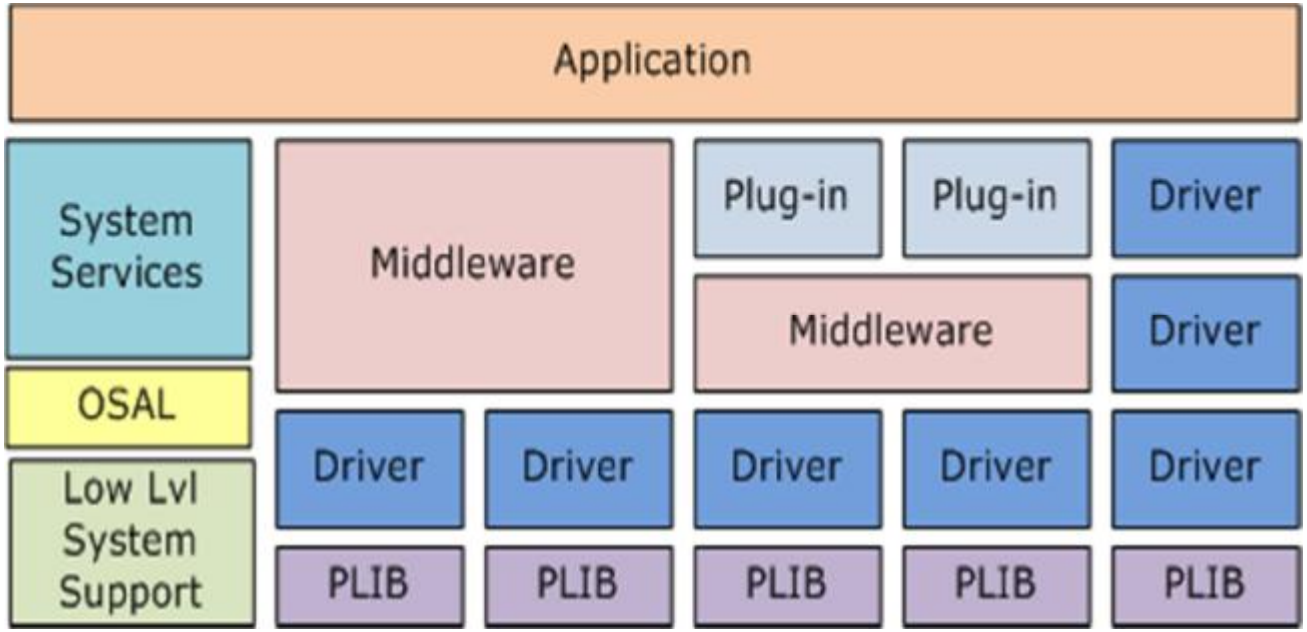
Что такое Harmony?

- Harmony – это кросс-контроллерный полностью интегрированный программный стек, «дружественный» к RTOS.
- Обеспечивает простые абстрактные интерфейсы по управлению периферией всей гаммы контроллеров Microchip

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Архитектура

- Поддерживает интеграцию библиотек приложений Microchip’s
- Фреймворк для разработки будущих библиотек
- Определяет абстрактный уровень операционных систем для простой интеграции с RTOS

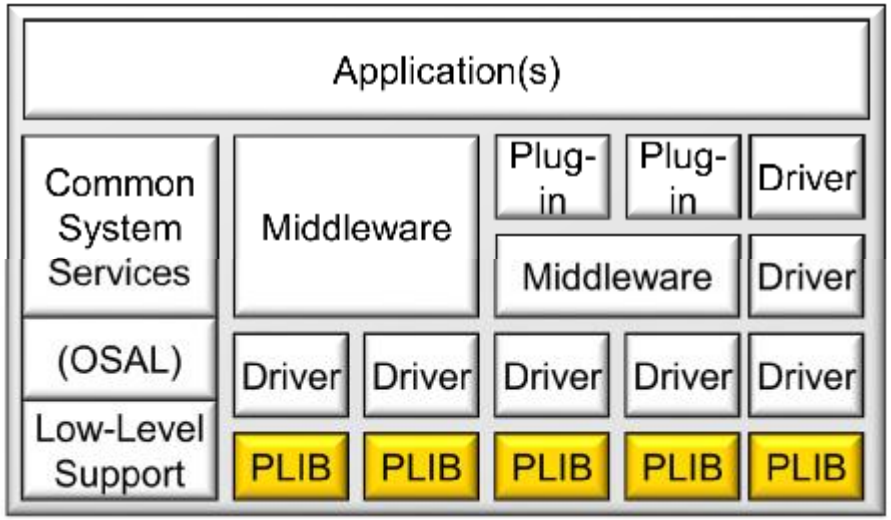


© 2013 Microchip Technology Incorporated. All Rights Reserved.

Библиотеки периферии

Функции

- **Функциональный интерфейс**
  - Простейшие операции
  - Реализует все возможности
  - Работает с битами и регистрами
- **Совместимость**
  - Единый интерфейс для всех микроконтроллеров
  - Обеспечивает непосредственный доступ к аппаратной части
- **Индексируемые экземпляры аппаратной периферии**
- **Реализация зависит от микроконтроллера**
  - Макросы без использования состояний (Stateless)
  - Нет блокирующих операций



Особенности:

- **Выглядит как вызов функций**
- **Без эмуляции**

  Если в аппаратной части возможность отсутствует, то ее интерфейс не поддерживается
- **Без контроля доступа**
- **Без управления состояниями**

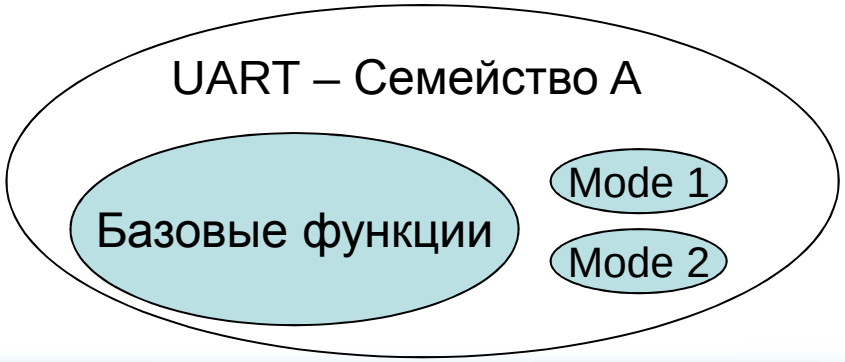
© 2013 Microchip Technology Incorporated. All Rights Reserved.

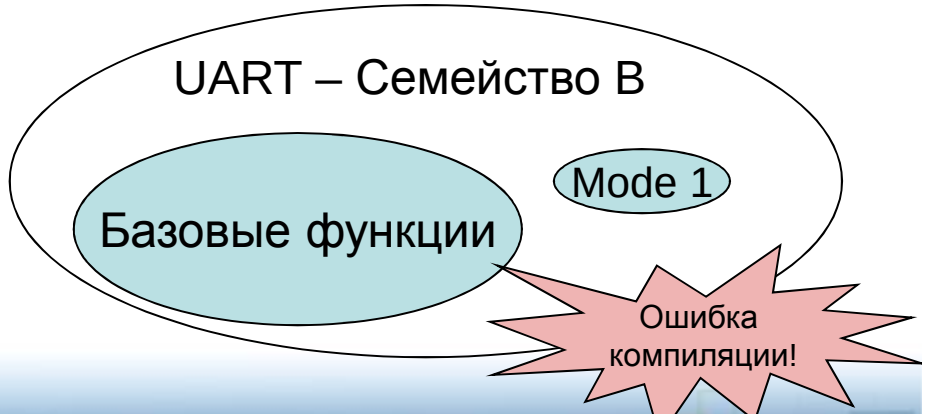
Библиотеки периферии

Наборы API: Базовые и дополнительные функции

```
// Базовые функции
PLIB_UART_Enable();
PLIB_UART_Disable();
PLIB_UART_DataSend(data);
data = PLIB_UART_DataReceive();
status = PLIB_UART_StatusGet();
PLIB_UART_StatusClear(status);
```

```
// Поддержка Mode 1
PLIB_UART_Mode1Enable();
PLIB_UART_Mode1Disable();

// Поддержка Mode 2
PLIB_UART_Mode2Enable();
PLIB_UART_Mode2Disable();
```



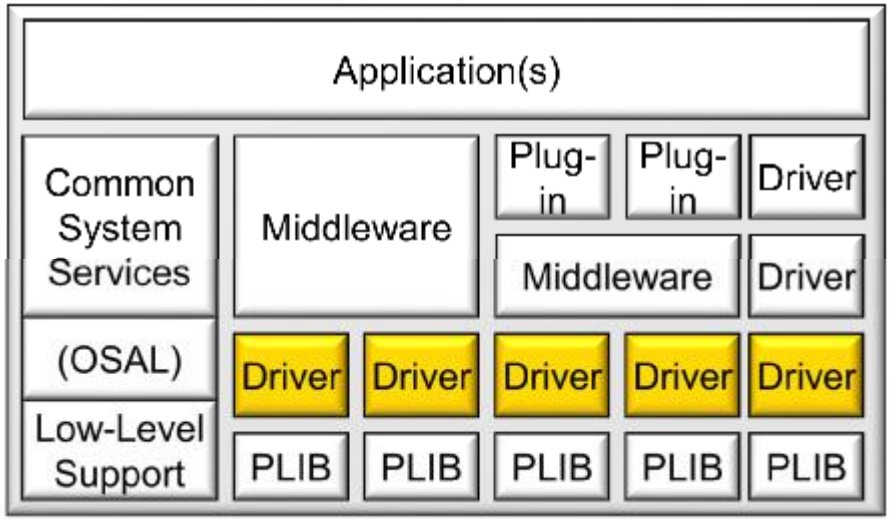


© 2013 Microchip Technology Incorporated. All Rights Reserved.

Драйвера устройств

Функции

- **Простой API**
  - Initialize / Deinitialize
  - Open / Close
  - Read / Write (при необходимости)
  - Device-Specific Functions
- **Машина состояний**
- **Множественные клиенты**
- **Множественные экземпляры HW**
- **Доступ к HW через:**
  - PLIB – для собственной HW
  - Сервисы SYS для совместно используемой HW
  - Другие драйвера для другой HW
- **Многопоточность (thread-safe) через OSAL**

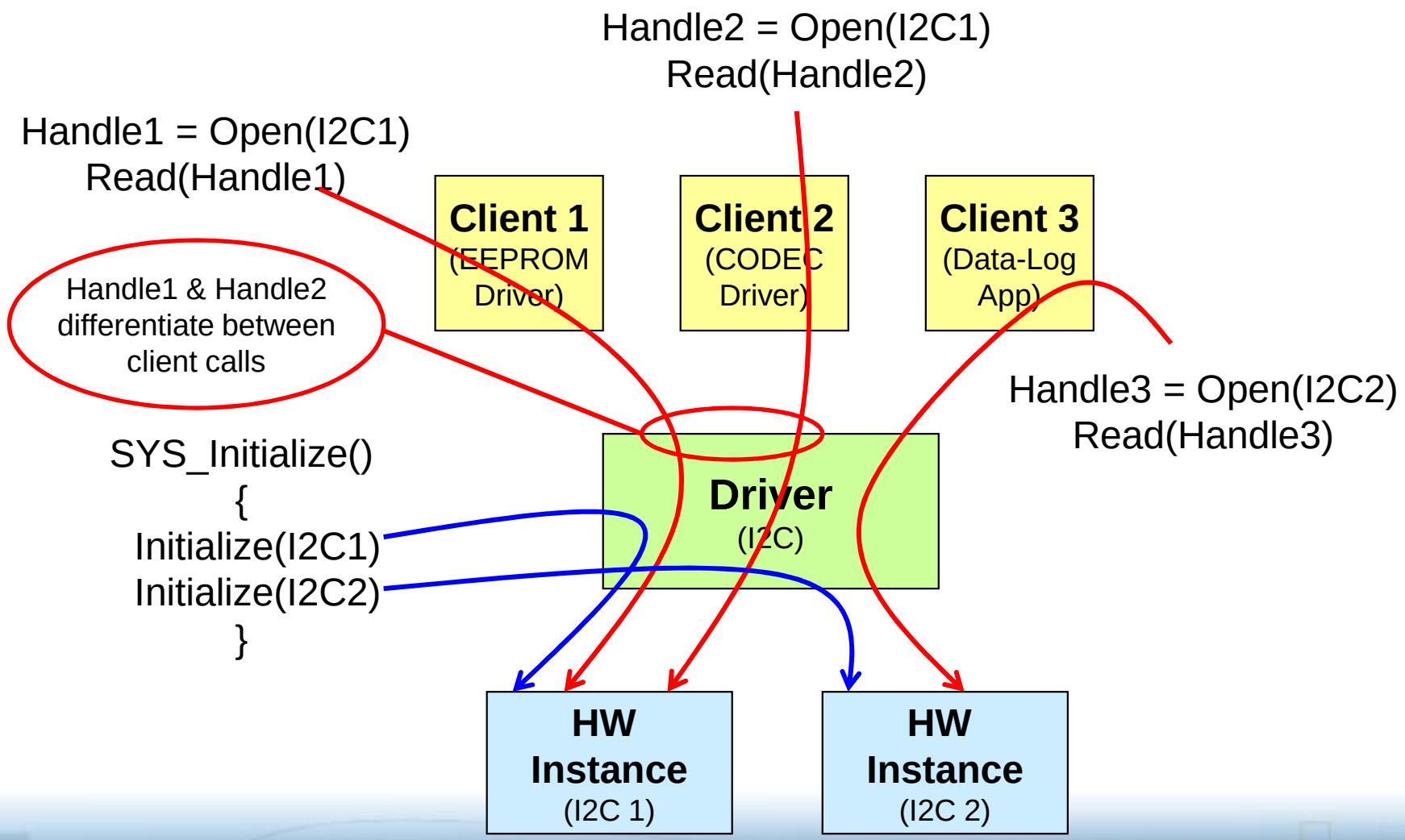


Конфигурации

- **Поллинг или прерывания**
- **Статические или динамические**
- **Блокирующие или не блокирующие**
- **Эксклюзивный или разделяемый доступ**
- **Дополнительные возможности**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Драйвера устройств
(Multi-HW, Multi-Client)

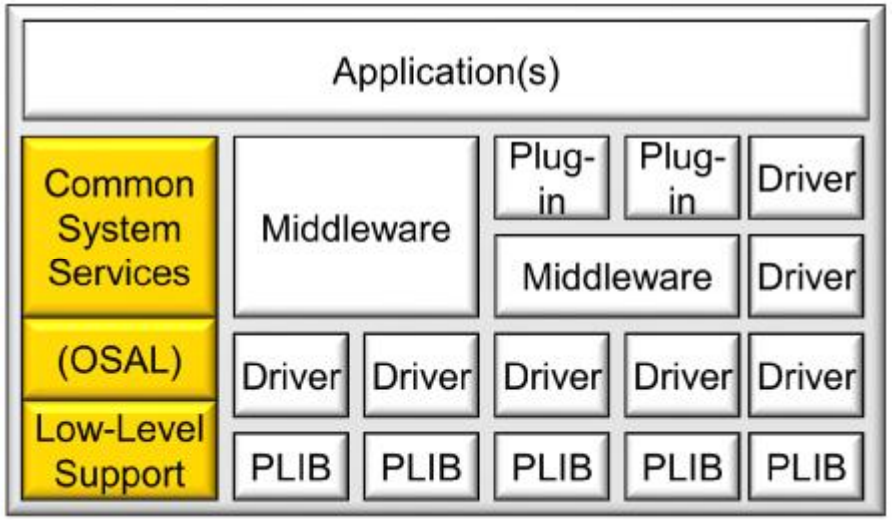


© 2013 Microchip Technology Incorporated. All Rights Reserved.

Системные службы

Функции

- **Общая функциональность**
  - Исключает дублирование и конфликты
- **Совместно используемые ресурсы**
  - OSC/Clock Control
  - Interrupt Control
  - Peripheral Pin Selection
  - System Timers
  - Debug Output
- **Абстрактный уровень OS**
  - Интерфейс к OS
  - Поддержка Thread Safety (Semaphores, Mutexes, и ..д.)
  - «Нулевая» реализация для систем без OS



Низкоуровневая поддержка

- **Поддержка системного уровня**
  - Инициализация системы
  - ISR
- **MCU-зависимые функции**
- **Конфигурация внешней аппаратуры**
  - Board support package

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Сервисы OSAL

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Библиотеки приложений

Функции

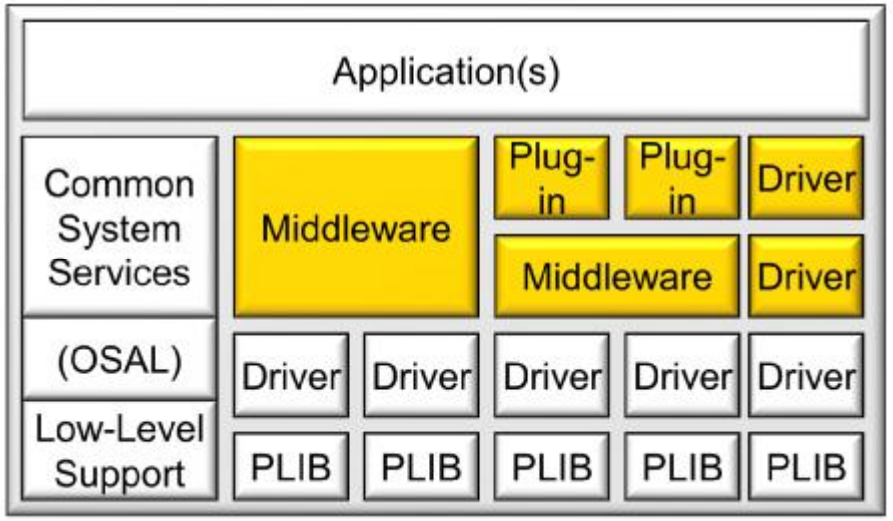
- **Реализуют требуемую функциональность**

  Сверх того, что поддерживается непосредственно HW

- **Обеспечивают абстрактный API**
- **Без непосредственного доступа к HW**

  Использует драйвера, системные службы, OSAL

- **Модульная архитектура**
- **Thread Safe/RTOS Ready**
  - С блокировкой, при наличии RTOS
  - Без блокировки, при отсутствии OS



Различные формы

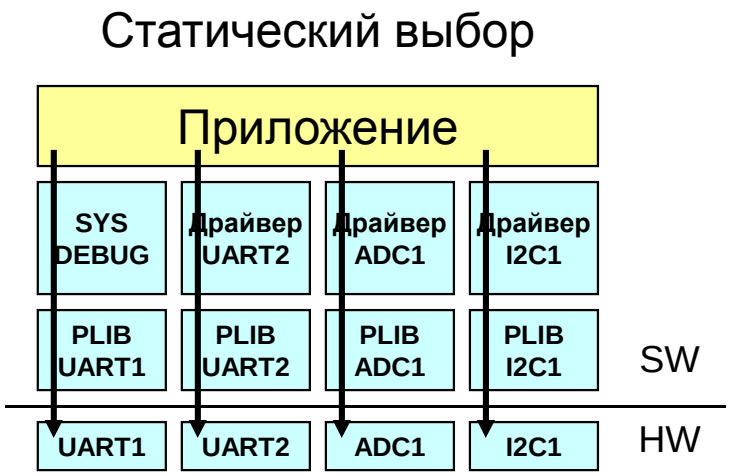
- Простые – без дополнительных функций
- Модули с поддержкой управления потреблением
- Полностью абстрактные драйвера
- Сложные стеки протоколов
- Могут включать “Plug-In” уровни

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Статические библиотеки

Статический выбор используемого экземпляра HW

- Выбор периферии на этапе компиляции
- Каждый вызов периферии из приложения всегда связан с конкретной периферией
- Закрывает 80-90% случаев
- Малая избыточность
- Несколько копий
  - Исходного кода
  - Объектного кода



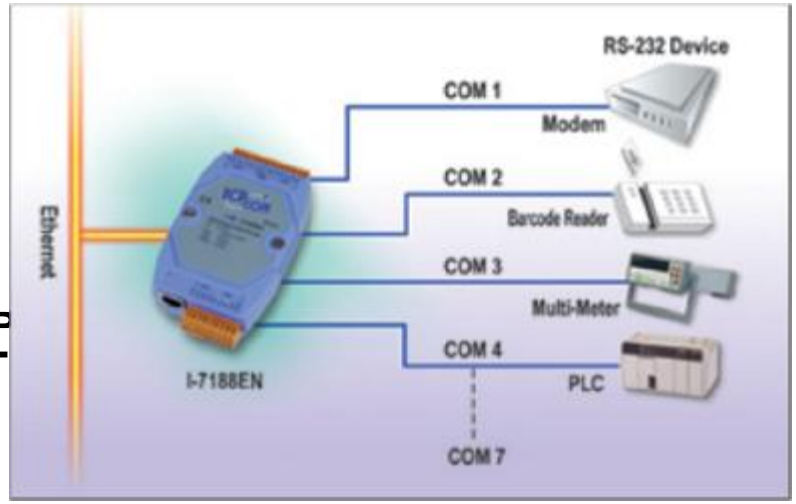
© 2013 Microchip Technology Incorporated. All Rights Reserved.

Динамические библиотеки

Динамический выбор используемого экземпляра HW

- **Периферия выбирается “динамически” во время выполнения**

  Выбор определяется на базе вычисляемой (или принимаемой) информации
- **Разработчик не может предсказать какой экземпляр периферии будет выбран**
- **Требует избыточности**
- **Потенциальные преимущества:**
  - Упрощает разработку и поддержку (Один экземпляр исходного кода)
  - Может уменьшить размер приложения (Один экземпляр объектного кода)
  - Более простая модель использования (Передача индекса в одну библиотечную программу вместо вызова разных подпрограмм)

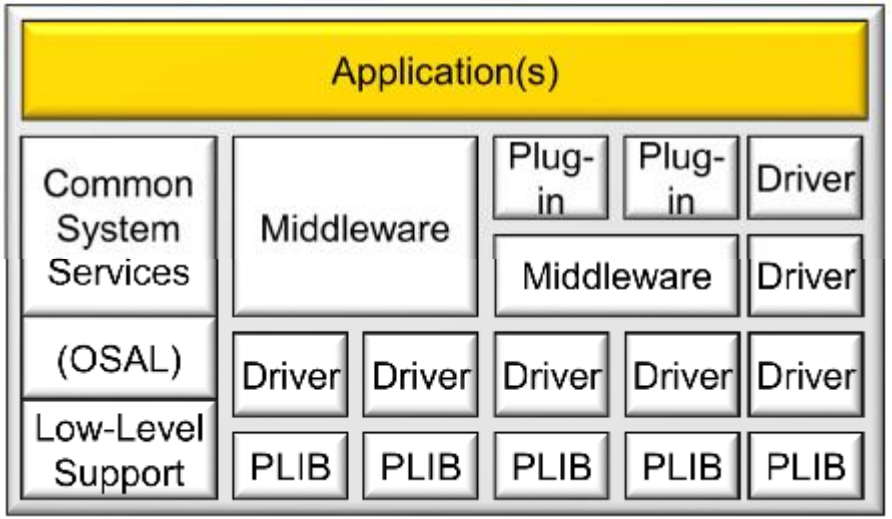


© 2013 Microchip Technology Incorporated. All Rights Reserved.

Уровень приложения

Функции

- **Реализует заданное поведение**
- **Легко мигрировать на другие uC**
  - Нет прямого доступа к HW
  - Использует драйвера, библиотеки приложений, системные службы
- **Выбор использования RTOS**
  - Возможна многопоточность
  - Несколько приложений
- **Определение системы**
  - Конфигурация
  - Инициализация
  - Машина состояний



Типы приложений

- Без OS, поллинг
- Без OS, прерывания
- С OS , многопоточное
- Конфигурируемое (на любое из перечисленных)

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Темы

- Недостатки традиционного подхода
- Основные концепции нового подхода
  - Переносимость (Portability)
  - Конфигурируемость (Configurability)
  - Модульность (Modularity)
  - Совместимость (Compatibility)
  - Гибкость (Flexibility)
- Введение в Harmony
- **Заключение**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Основные преимущества

- **Улучшенная совместимость для разных микроконтроллеров**
  - Общий, одинаковый API
  - Легко расширяется на более мощные микроконтроллеры
  - Легко сужается на менее мощные микроконтроллеры
- **Улучшенное взаимодействие модулей**
  - Драйвера и библиотеки легко компонуются вместе
  - Легкий перенос на другую аппаратную платформу
- **Ускоренный выход на рынок**
  - Возможность быстро разрабатывать приложения
  - Возможность быстро добавлять новые функции
- **Облегчение работы с библиотеками**
  - Исключение конфликтов
  - Улучшенное качество кода (Процесс выпуска версий)

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Процесс выпуска версий

Alpha Release

- Принципы определены, основные функции реализованы
- Протестированные модули
- Протестированная компиляция

Beta Release

- Интерфейсы определены и подписаны
- Исправлены ошибки Alpha release
- Функционально протестировано

1.0+ Release

- Код определен и подписан
- Исправлены ошибки Beta Release
- Протестировано в режиме стресс-тес

Стабильный API
Стабильный код

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Экосистема

- **Поддержка RTOS**

- **Сторонние поставщики ПО**

© 2013 Microchip Technology Incorporated. All Rights Reserved.

Поддерживаемые RTOS

Планируемое состояние. Может быть изменено.

	FreeRTOS™	OpenRTOS™	Micrium uCOSII	Micrium uCOSIII
Бесплатно	ü			
Платная опция через Microchip Direct		ü	ü	ü
Поддержка Microchip		ü[1]	ü[1]	ü[1]
Поставляется вместе с MPLAB Harmony	ü		ü	ü
Примечание:		100% API совместимое с FreeRTOS	Бесплатная пробная версия в MPLAB Harmony	Бесплатная пробная версия в MPLAB Harmony

1. При покупке пользователем лицензии

© 2013 Microchip Technology Incorporated. All Rights Reserved.

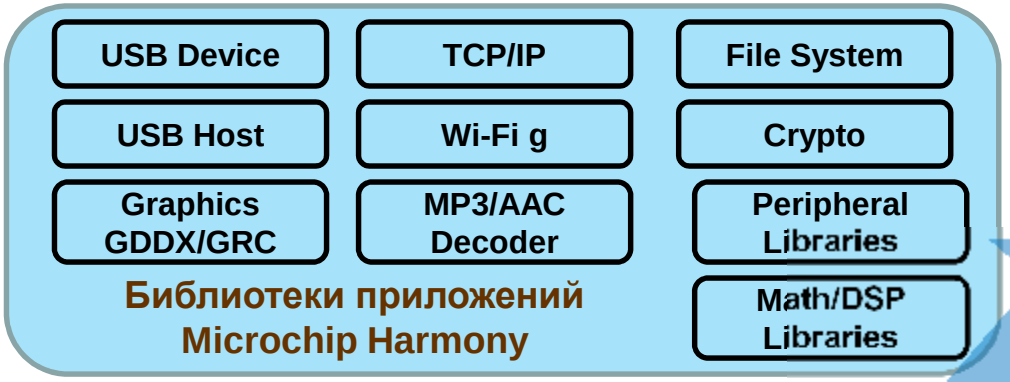
Экосистема

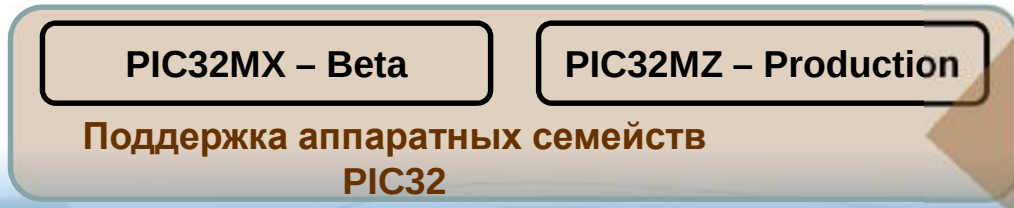
- **Основные возможности платформы:**
  - Модульные драйверы периферии и библиотеки приложений
  - Поддержка динамических драйверов с множественными клиентами
  - Поддержка RTOS
  - Масштабируемость и совместимость для различных микроконтроллеров Microchip
  - Встроенные, проверенные и поддерживаемые решения от сторонних поставщиков

© 2013 Microchip Technology Incorporated. All Rights Reserved.

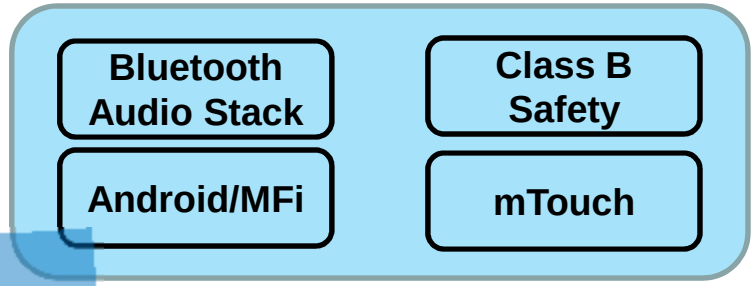
Развитие MPLAB Harmony

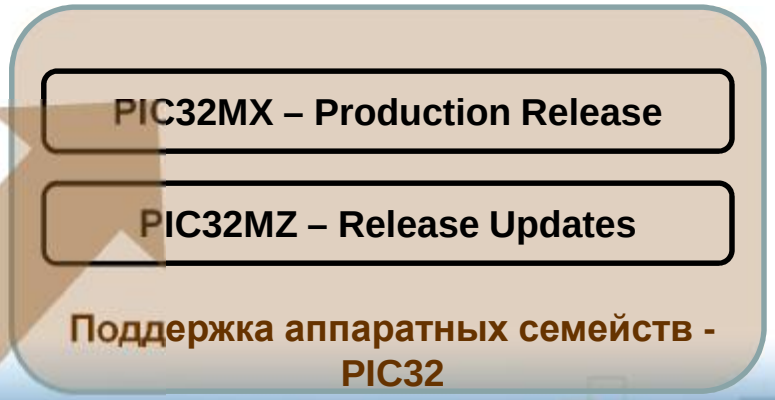
MPLAB Harmony v1.00 – 18 Nov ‘13





Harmony – ближайшее будущее





© 2013 Microchip Technology Incorporated. All Rights Reserved.

Спасибо за внимание !!!

Вопросы ?

© 2013 Microchip Technology Incorporated. All Rights Reserved.