Design Examples and Case Studies of Program Modeling and Programming with RTOS-2:

Lesson-4

CASE STUDY OF A MOBILE PHONE SOFTWARE FOR KEY INPUTS FOR CREATING SMSes

1. SMS Create and Send Application Requirements

Purpose

- To create an SMS message and communicate using a T9 format keypad and using the tasks for adding number, editing message and then sending the message

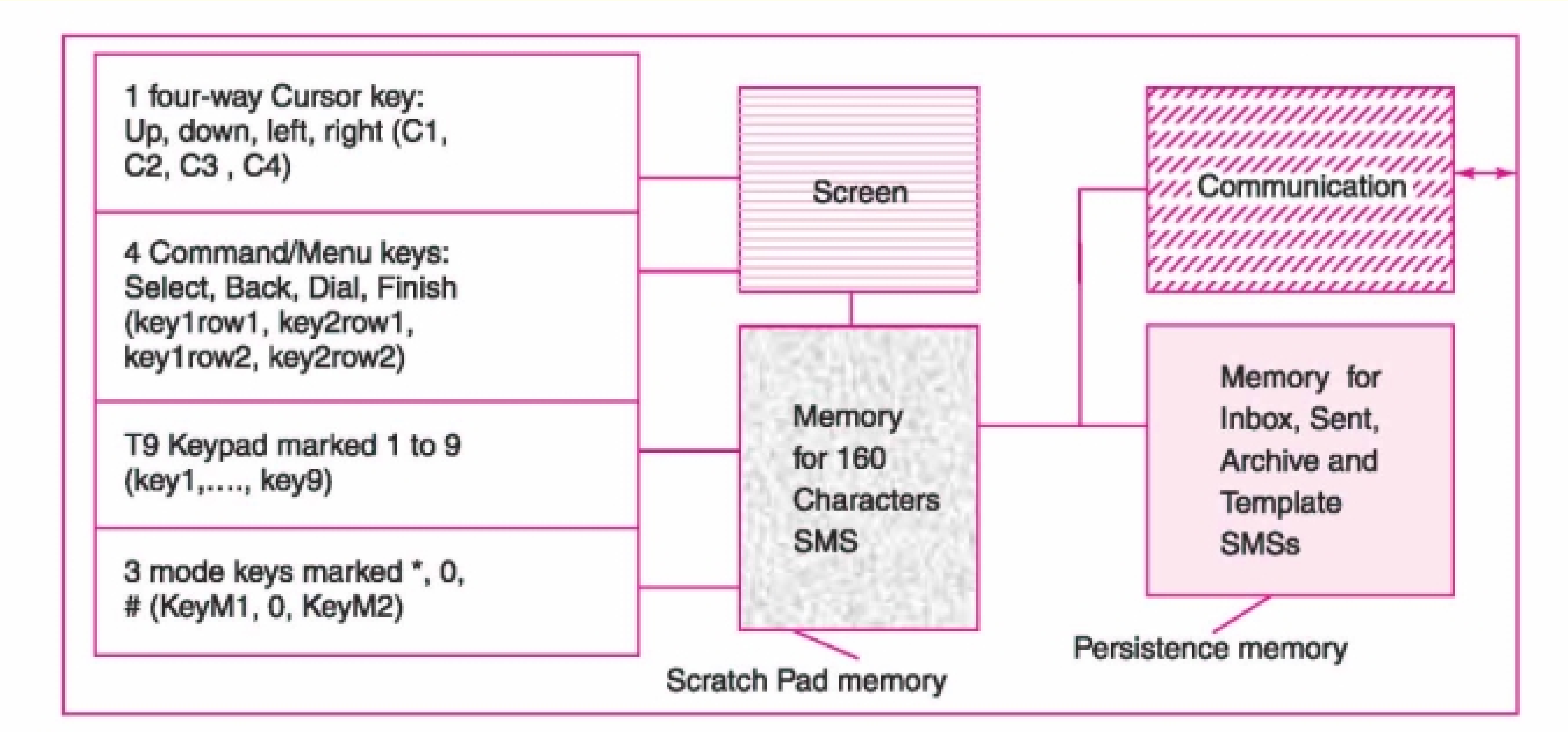
Inputs

- State of a T9 key (key1,…., key9) .
- State of mode key (keyM1 and keyM2)– marked * and #, which defines the functioning mode.

Inputs…

- State for command from one of the four command keys (key1row1, key2row1, key1row2 and key2row2).
- State of cursor input on the cursor clicks on move up, down, left or right using C1key, C2key, C3key or C4key.

Keypad, screen, memory and communication units


1 four-way Cursor key: Up, down, left, right (C1, C2, C3 , C4)
4 Command/Menu keys: Select, Back, Dial, Finish (key1row1, key2row1, key1row2, key2row2)
T9 Keypad marked 1 to 9 (key1,...., key9)
3 mode keys marked *, 0, # (KeyM1, 0, KeyM2)
Screen
Memory for 160 Characters SMS
Scratch Pad memory
Communication
Memory for Inbox, Sent, Archive and Template SMSs
Persistence memory

GUIs

- GUI─ using display of screen menu or text and a cursor, command select key, and the cursor position change keys.
- The cursor position can be changed up, down, left and right by using C1, C2, C3 and C4.
- The cursor when points on the screen to a line of menu for a command, it shows that menu item with blue or other background.
- Cursor when points to a character in a line of text or phone number, it shows a vertical line at the right of the character position.

GUIs

- At the cursor position, a text can be selected or entered by a click of command select key (Key1Row1).
- At the cursor position, the shown character can be cleared by a click of clear or back key (Key2Row1).

Signals, Events and Notifications

- E_NewState and message for the *state* of the key post on interrupts from command key or any other key.

Signals, Events and Notifications…

- Event E_SMS, which starts on SMS_Create_Text application.
- Event E_SMS occurs on a set of four GUI notifications – *MsgMessages*, *MsgTextMessages*, *MsgCreate*   and *MsgText*.

Signals, Events and Notifications…

- *MsgMessages* notification on selecting option Messages,
- *MsgTextMessages* on selecting option Text_messages,
- *MsgCreate*   on selecting option Create_message, and
- Msg*Text* message on selecting option Text.]

Signals, Events and Notifications…

- Notifications for display on completion of tasks.
- For example, after sending SMS, display notification ‘Message sent’ and before completing the transmission of SMS ‘Sending message’.

Signals, Events and Notifications…

- ISR_T_Deacitvate— time-out interrupt switches off the phone at display of idle state (start up menu display in idle state) to the display off and sleep state.
- Interrupt after a no action for a period longer than a programmed period 15 s.
- ISR_T_Out_Help_Option — time-out interrupt when a cursor or marked menu displays an option for a period longer than the programmed period, 15s.

Outputs

- SMS_Create_Text string, to displayon the screen and also save in scratchpad memory during the editing and also saved in sent folder after sending the SMS
- Screen menu text lines for displaying option(s), text of menu, marked text or character to enable its selection by clicks.

Outputs

- Help menu text display to display action, which will take place on selecting an option after a T_Out_Help_Option interrupt. Option is assumed as one at which cursor points.

Scratchpad Memory

- For 160 characters maximum in an SMS message.

Persistence memory

- Persistence memory addresses (in flash memory) for SMSes in Inbox, Sent, Archive and Template.

Functions of the system

- When set of options consisting of 4 notifications, one for a command *Messages*, next for command-option *Text Messages*, next for application *Create Message* and last for *Text*, generates, an event E_SMS
- E_SMS signals creation of text messages entered and edited with T9 keypad of nine keys

Functions of the system

- When any key clicked by user, its state computed based upon key's earlier state, cursor present position and timer status and count
- Notification E_NewState is posted in a message box to send signal to an ISR, task or application for the required action.

Functions of the system…

- For choosing SMS creating text application, the cursor and command-cum-options select (Key1Row1) key are used as follows:
- (a) When command key (key1Row1) is used to select ‘command Messages’, a command-option is then selected using a GUI.
- (b) User selects *Text Messages* option by clicking for the displayed option ‘Text messages’.

Functions of the system…

- (c) A menu then shows up for selecting one of the followings using the cursor down and up keys: Create message, Inbox, Sent items, Archive, Templates, My folders, Distribution lists, Delete messages and Message settings.
- User selects ‘Create message’ for selecting SMS creation application.

Functions of the system…

- (d) A menu then shows up for selecting type of message to be created.
- Cursor can select one of the following menu items on display: Text and Numeric page.
- User selects ‘Text’ for creating SMS text.

E_SMS post

- E_SMS posts (signals) to start execution of SMS create and send application when user finishes selection of four options Messages, Text Messages, Create Message and Text.

On E_SMS post

- The tasks Add number, Edit Message and Send Message execute in order to specify (i) a new or additional mobile number for sending the SMS, (ii) editing the SMS message by keying in from T9 keypad and (iii) sending it over the given mobile, respectively.

State S_Wake

- When mobile is inactive, display screen startup display shows up on key2row2 interrupt, which causes an ISR_key2row2 routine execution and state of mobile becomes wake up state S_Wake.

State S_Sleep

- When mobile is in active state and is running an application, it is brought to idle state by at another key2row2 interrupt.
- Display screen switches off (shuts down) after a preset interval, say, 15 s and the state of mobile becomes sleeping state S_Sleep

Design metrics

- *Power Dissipation*: Battery operation
- *Performance*: 3 minute for 1 SMS message creation and send.
- *Engineering Cost*: US$ 20000 (assumed) for software
- *Manufacturing Cost*: None once the codes are ready and tested

Test and validation conditions

- All commands and options functioning tested.

2. Classes

Classes

- Orchestrator,
- Task_SMS_CreateTextSend
- Task_ScreenDispl.

Classes

- ORCHESTRATOR class– extends to Orchestrator_CommandsGUIs and to Orchestrator_SMSCreateSend.
- Task_Messages, Task_TextMessages, Task_CreateMessage and Task_Text and interface keypad interrupt ISR_KINT .
- Task_SMS_CreateTextSend extends to Task_AddNumber, Task_AddEmail, Task_AddList, Task_EditMessage, Task_ListRecipient and Task_ Send.

Screen displays Classes

- Start up screen display
- menu items display
- SMS display during editing task and action display during sending the SMS
- Task_ScreenDispl– extends to four classes Task_StartUpDispl, Task_ SMSDispl, Task_ActionDispl and Task_ MenuTextLinesDispl.

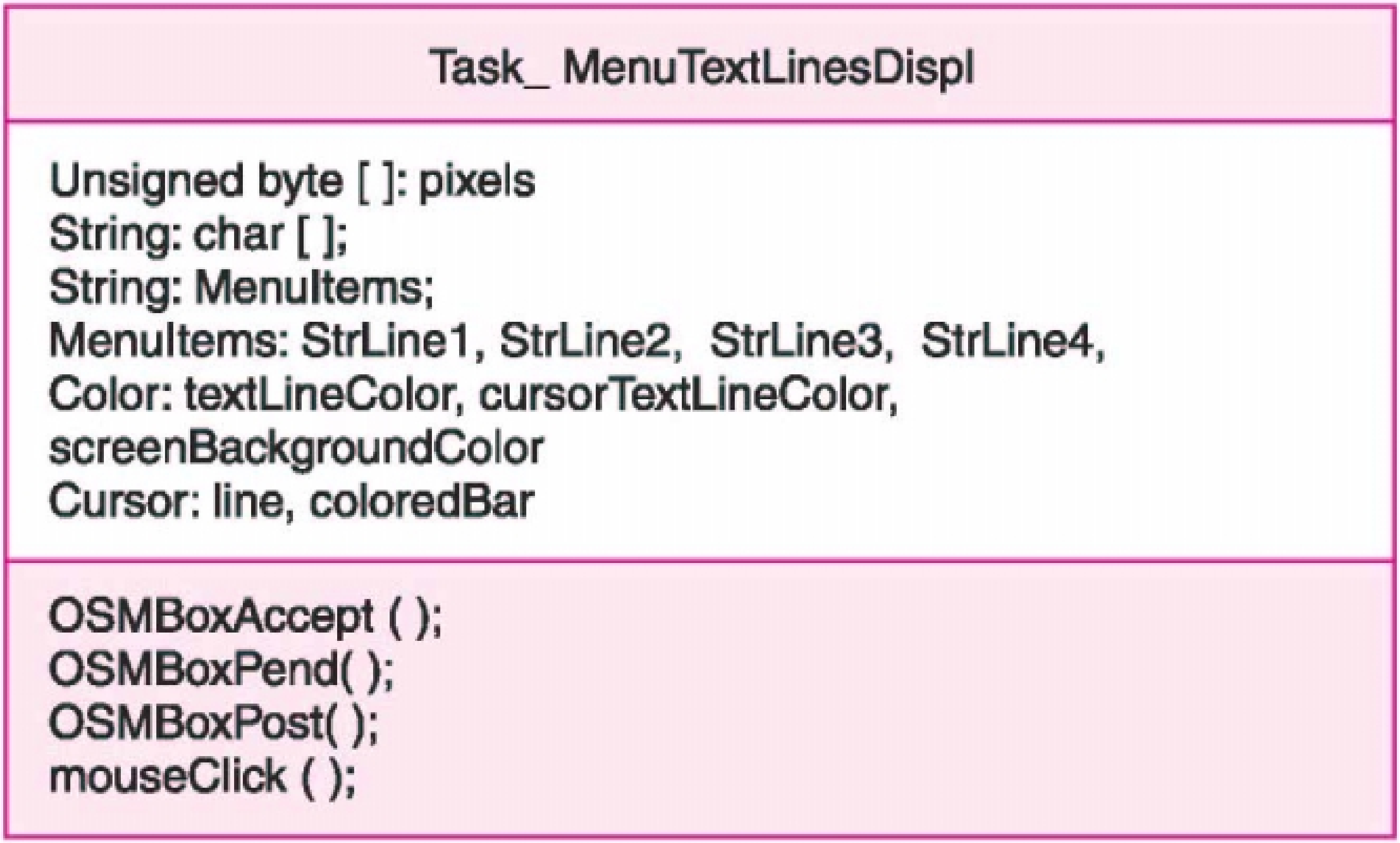
ISRs

- ISR_WirelessPort, ISR_T_Out_Help_Option, ISR_T_Deacitvate and ISR_KINT.
- ISR_KINT runs the service functions for any of the state transitions of twenty key and senates notifications for the state of a key Key1Row1 or Key1Row2 or , Key2Row1, Key2Row2, C1 to C4, M1 or M2 or keys 0 to 9.

Task_ MenuTextLinesDispl

- Pixels field of Unsigned byte [ ].
- A string– an array of characters.
- StrLine1, StrLine2, StrLine3 and StrLine4 are the strings in the object of MenuItems.
- The color fields are textLineColor, cursorTextLineColor, screenBackgroundColor.
- Cursor has two fields line and colored bar. The methods are OSMBoxAccept ( ); OSMBoxPend( ); OSMBoxPost( ) and mouseClick ( ).

Class Task_ MenuTextLinesDispl



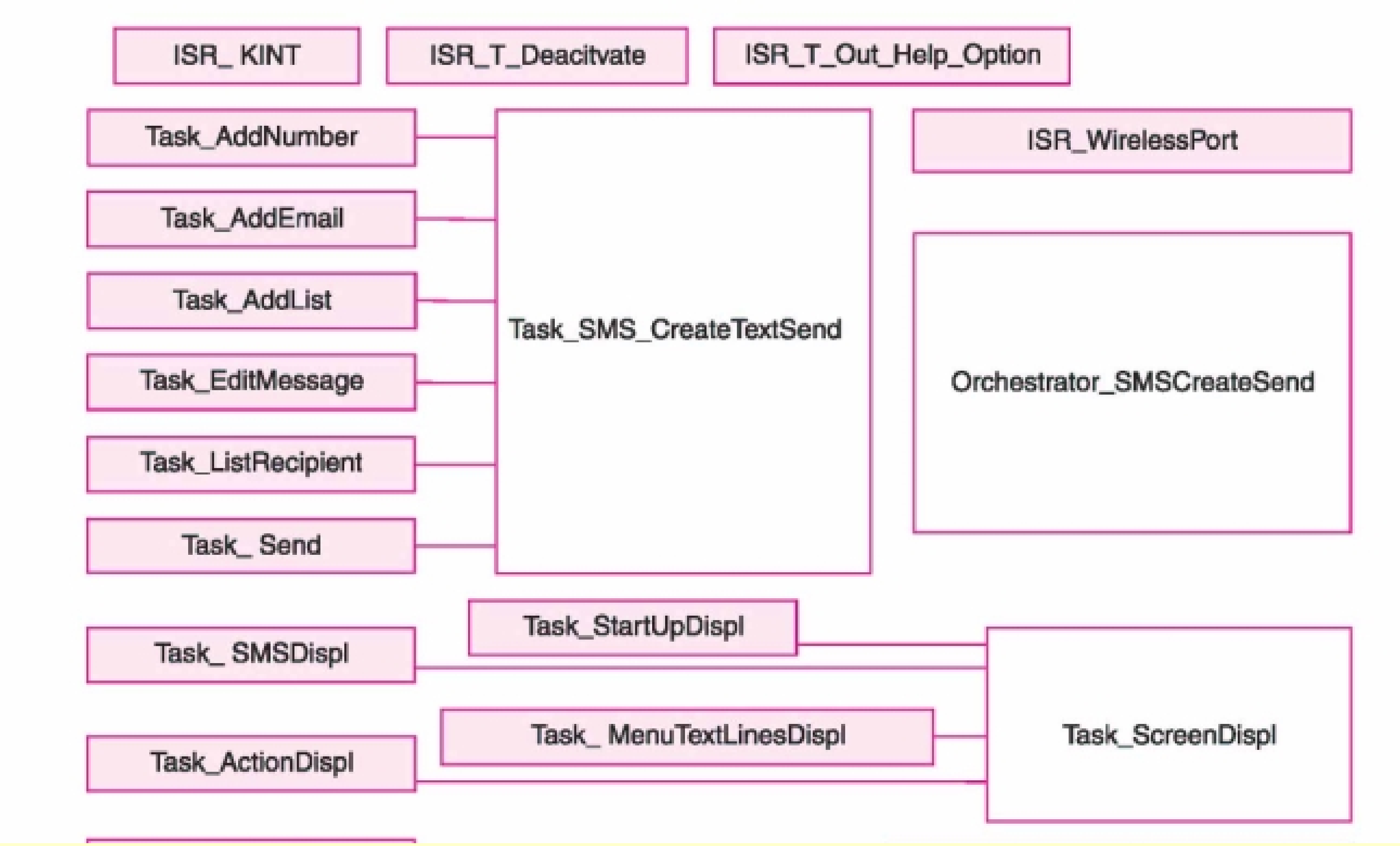
Task_ MenuTextLinesDispl

Unsigned byte [ ]: pixels
String: char [ ];
String: MenuItems;
MenuItems: StrLine1, StrLine2, StrLine3, StrLine4,
Color: textLineColor, cursorTextLineColor,
screenBackgroundColor
Cursor: line, coloredBar

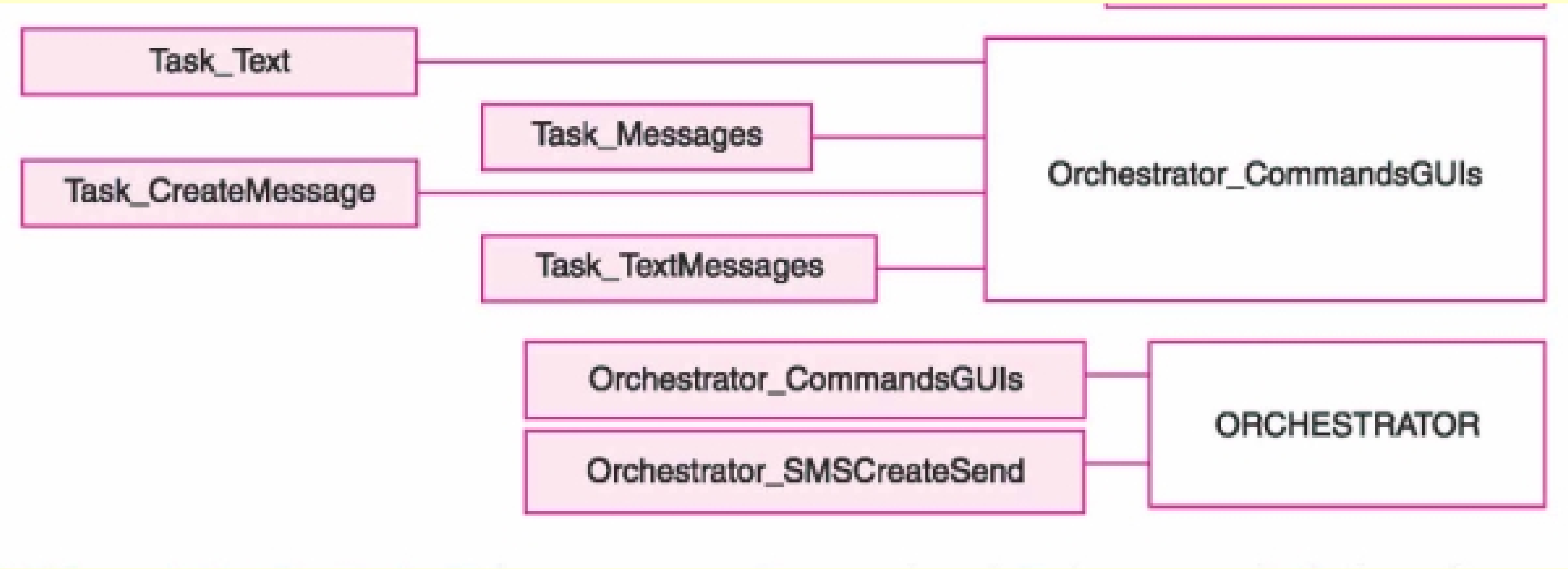
OSMBoxAccept ( );
OSMBoxPend( );
OSMBoxPost( );
mouseClick ( );

3. Class diagrams

Class diagrams



Class Diagrams


Task_Text
Task_Messages
Task_CreateMessage
Orchestrator_CommandsGUIs
Task_TextMessages
Orchestrator_CommandsGUIs
ORCHESTRATOR
Orchestrator_SMSCreateSend

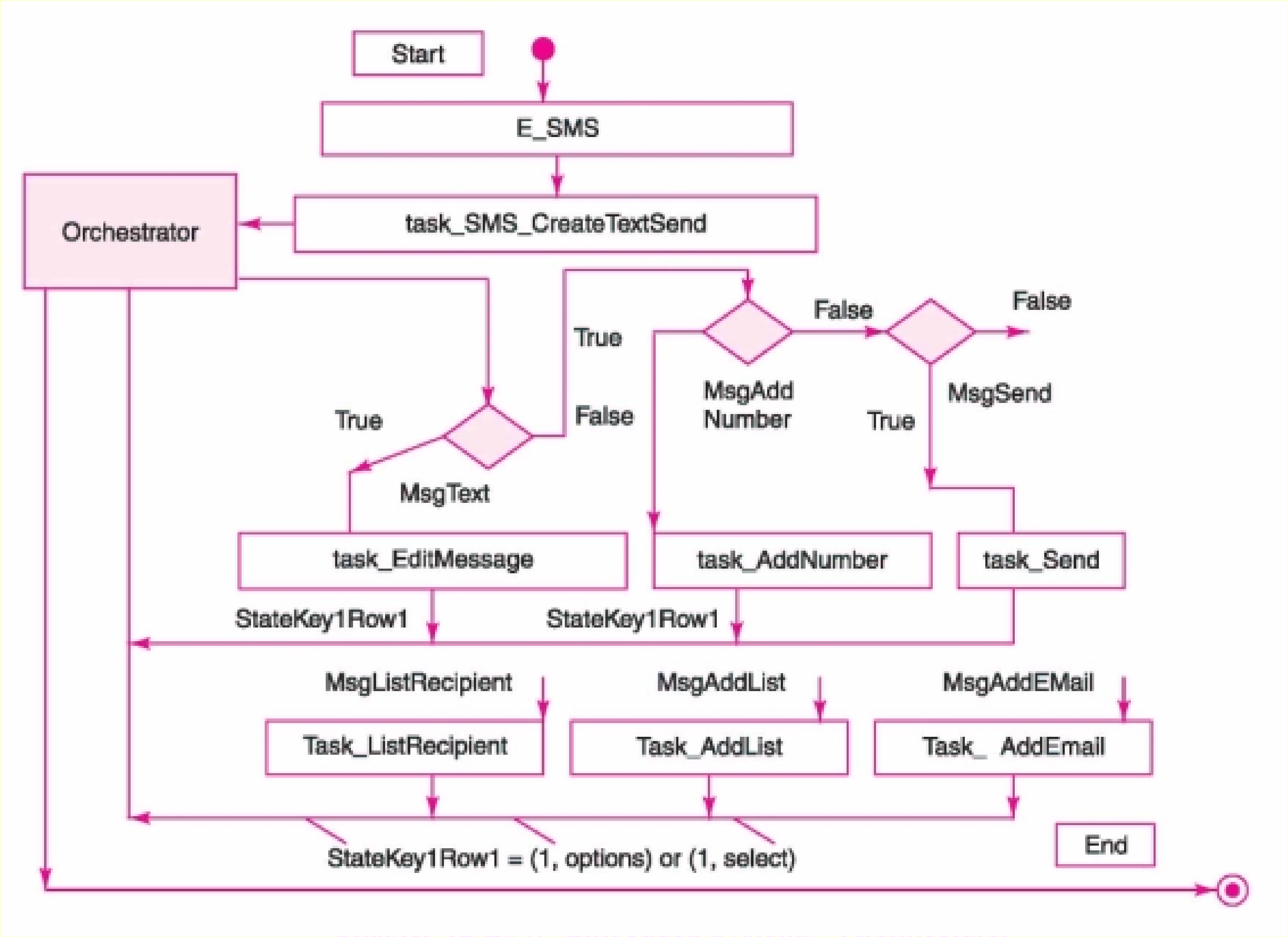
4. Message queue Objects

Message queue Objects

- For posting to the ISRs and tasks.
- Message queue objects are accepted or waited for at the tasks
- *MsgMessages*, *MsgTextMessages*, *MsgCreate* and *MsgText* are posted on Key1Row1 interrupts.
- Event objects are posted on a set of notifications
- E_SMS is posted on *MsgMessages*, *MsgTextMessages*, *MsgCreate* and *MsgText*.
- E_NewState is posted on any new state generation from any keys in the mobile

5. State Diagram

State diagram for task_SMS_CreateTextSend



State transitions

- Between the tasks, task_SMS_CreateTextSend and task_AddNumber, task_AddEmail, task_AddList, task_EditMessage, task_ListRecipient and task_ Send.
- A sate transition each occurs after notification of MsgAddNumber, MsgAddEmail, MsgAddList, MsgEditMessage, MsgListRecipient and MsgSend on selection of menuItems Add Number, Add Email, Add List, Edit Message, List Recipient and Send, respectively.
- Task_Send

6. Hardware Architecture

Hardware architecture

- Same as mobile phone system with SMS features
- T9 keypad
- Processors, ASIPs, keys, memory, ports and devices
- Interfacing and mapping of these components.

Command keys for SMS keying in

- Four number *command keys*, key1row1, key2row2, key1row2 and key2row2 are present. Use of Key1Row1 and Key2Row1 are similar to left and right clicks in computer-mice for GUIs.
- Use of Key1Row2 and Key2Row2 clicks are similar to click to *start* menu item and *turn-off* or restart menu item, respectively, in a computer start up window

Cursor keys for SMS keying in

- Four number *cursor keys*, which are pressed to move the cursor up, down or left or right of character when editing the SMS when it is being created. Their actions are similar to ↑, ↓, ← and → keys in a key board.
- On cursor key interrupt on clicks, the notifications are sent for the states of keys C1key, C2key, C3key and C4key and current cursor display position.

T9 Keys

- T9 keypad is used for keying in of SMS messages. T9 keypad has nine keys 1 to 9 3. T9 key (key1,…., key9 key) inputs are used for dialing numbers as well as editing the text inputs during SMS create application

Mode keys

- Two keys marked * and # key notify the states of keyM1, key0 and keyM2
- To protect accidental use of the keys when phone is kept in the pocket, the key1row1 and M1 are simultaneously pressed, the pad undergoes transition to lock state if previously it was in unlock state and to unlock state if it was in lock state.
- Bilingual text SMS editing. M2 is used to convert the English text mode to other language text mode.

Display Screen

- Screen displays and GUIs for start up display, menus and options, text being edited or actions currently taking place.

7. Software Architecture

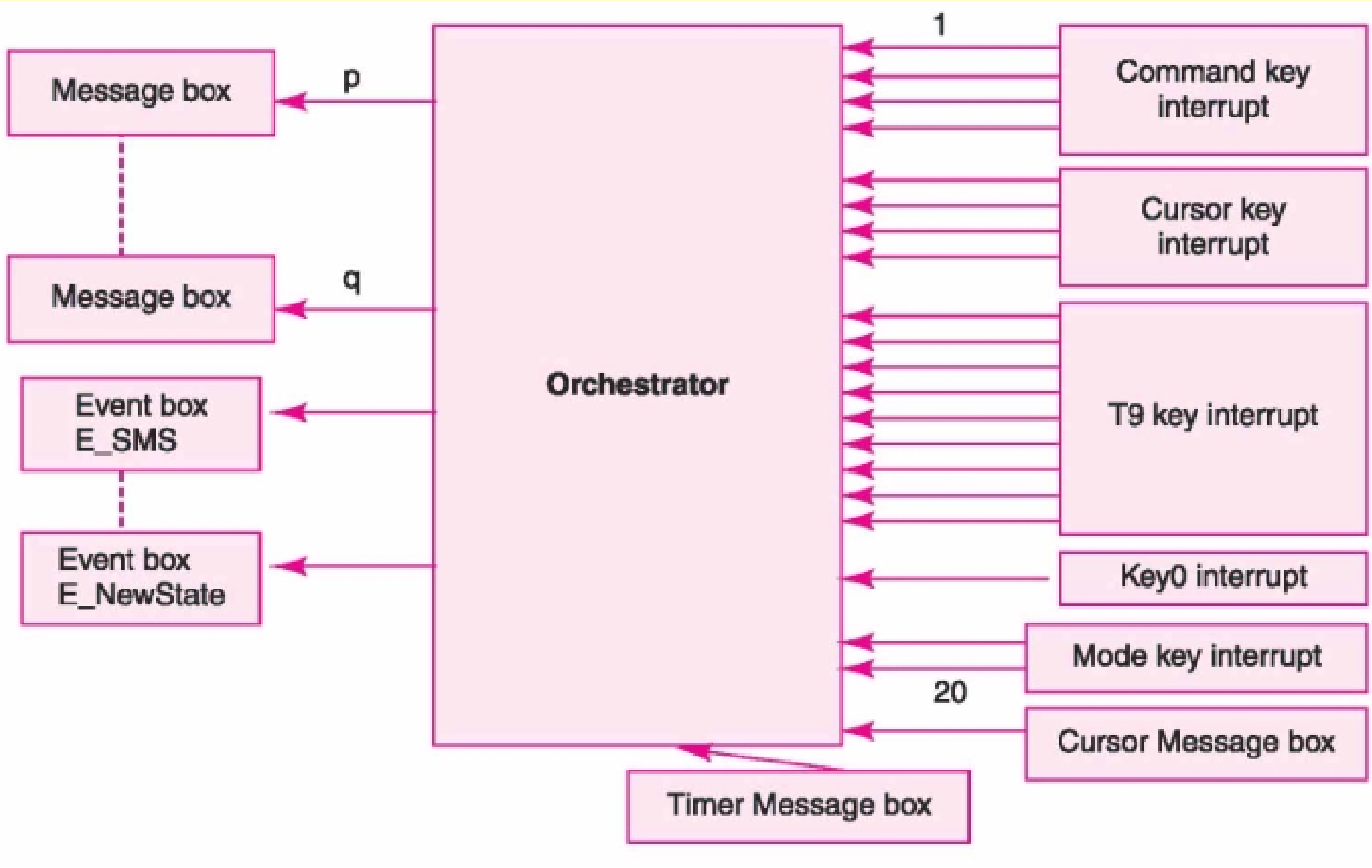
Software architecture for the SMS create and send application

- OS controls the OSMsgAccept, OSMsgQPost and OSMsgPend message queue functions at message boxes and event functions at the even boxes.
- OS synchronizes such that tasks and facilities concurrent processing of SMS application on the mobile.
- Key-system layer using Orchestrator ISR_KINT
- Application layer

Key-system layer

- A key click generates an interrupt and a service routine ISR_KINT, which then executes an Orchestrator.
- The ISR_KINT reads the port status bits to find which key has been clicked, also read the timer status and timer counts and cursor position and that position menu or text message.
- It signals the Orchestrator to initiate and generates the notifications and events and posts these into the message-boxes and event boxes for waiting tasks.

Key-system layer with ISR_KINT and Orchestrator in software architecture of mobile



ISR_KINT Orchestrator messages

- *MsgMessages* when cursor on display screen points to a Command_Msg Messages
- *MsgTextMessages* when cursor on display screen points to a command_option_Msg TextMessages,

ISR_KINT Orchestrator messages

- MsgCreate when cursor on display screen points to an application-option_Msg point to Create, [Screen displays the application options *Create message*, *Inbox*, *Sent items*, *Archive*, *Templates*, *My folders*, *Distribution lists*, *Delete messages* and *Message settings* options]
- MsgTextType when cursor on display screen points to a typeoption_Msg Text.

States generated from the key interrupts

- S_key0, S_Key1, S_Key2, …., or S_Key9 state as per the timer status and counts if there is new state of key0, key1, key2, …, key9.
- S_C1, S_C2, S_C3, or S_C4 state if C1 or C2 or C3 or C4 is clicked.
- S_M1 or S_M2 if M1 or M2 is clicked.
- S_key1Row1, S_key1Row2, S_key1Row2 or S_key2Row2 S_key1Row1 if a command key is pressed is clicked.

Events

- E_SMS if command, command option, application option and type options *MsgMessages*, *MsgTextmessages*, *MsgCreate* and *MsgTextType* were posted in step 1, 2, 3 and 4. [*MsgMessages*, *MsgTextMessages*, *MsgCreate* and *MsgTextType* initiates event object E_SMS.]
- E_NewState if any state change

Application layer

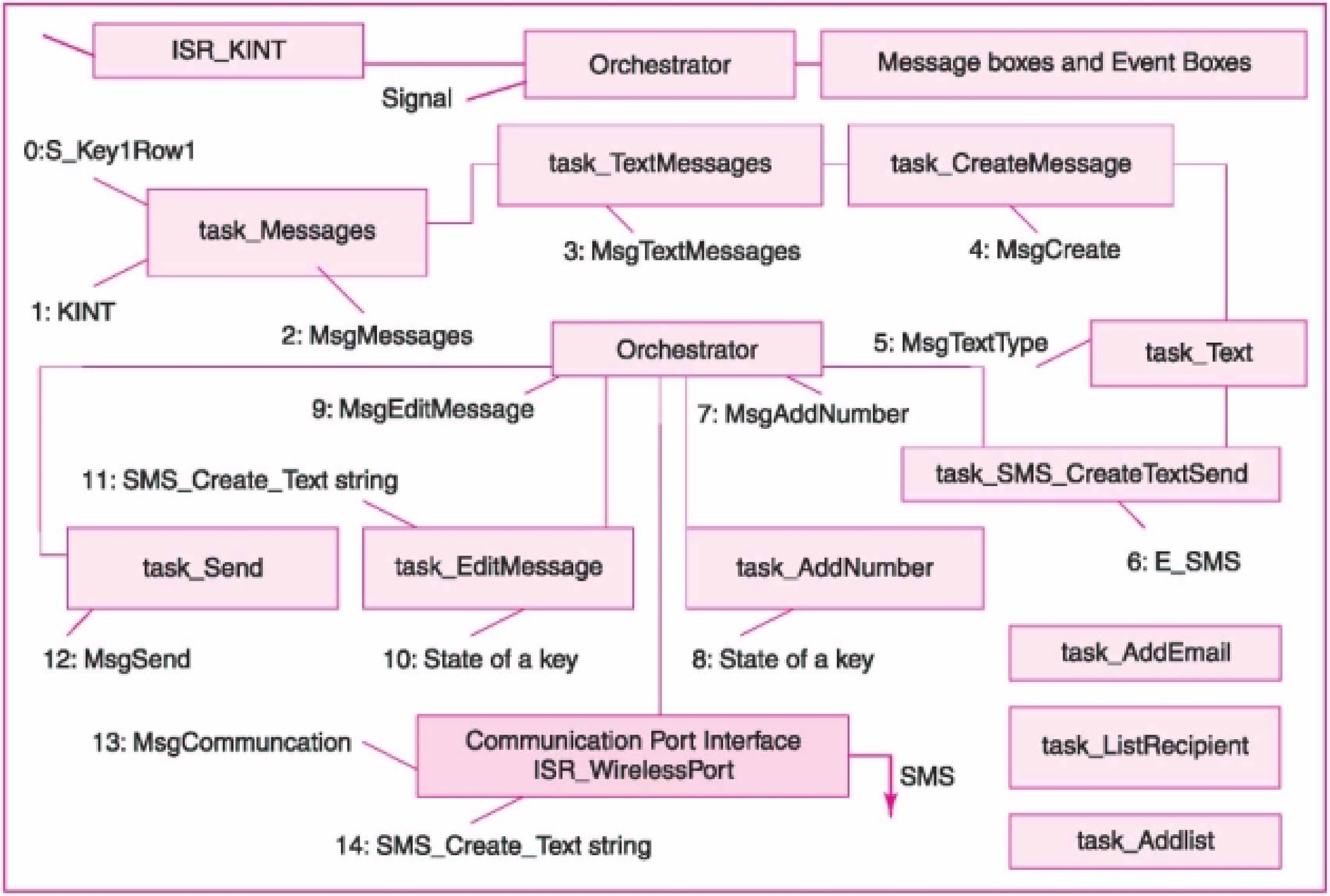
- Tasks
- ISRs for initiating action on user inputs and GUI notification

Application layer

- Task MenuTextLinesDispl.
- It executes on posting of a message object MsgTextMessages or MsgMessages or MsgTextMessages, MsgCreate   or MsgTextType into a message box by ISR_KINT
- Task_SMS_CreateTextSend for SMS create and send application and it executes on event E_SMS.

8. Multiple tasks and their synchronization model

Synchronization model for SMS create and application tasks, ISRs and Orchestrator tasks



Summary

We learnt

- SMS message creation and communication
- Uses of command keys, cursor keys and GUIs in mobile phone SMS create and send application.
- Concept of Orchestrator used in robot applications is also useful for the present mobile phone applications

End of Lesson-4 of chapter 12 on CASE STUDY OF A MOBILE PHONE SOFTWARE FOR KEY INPUTS FOR CREATING SMSes