

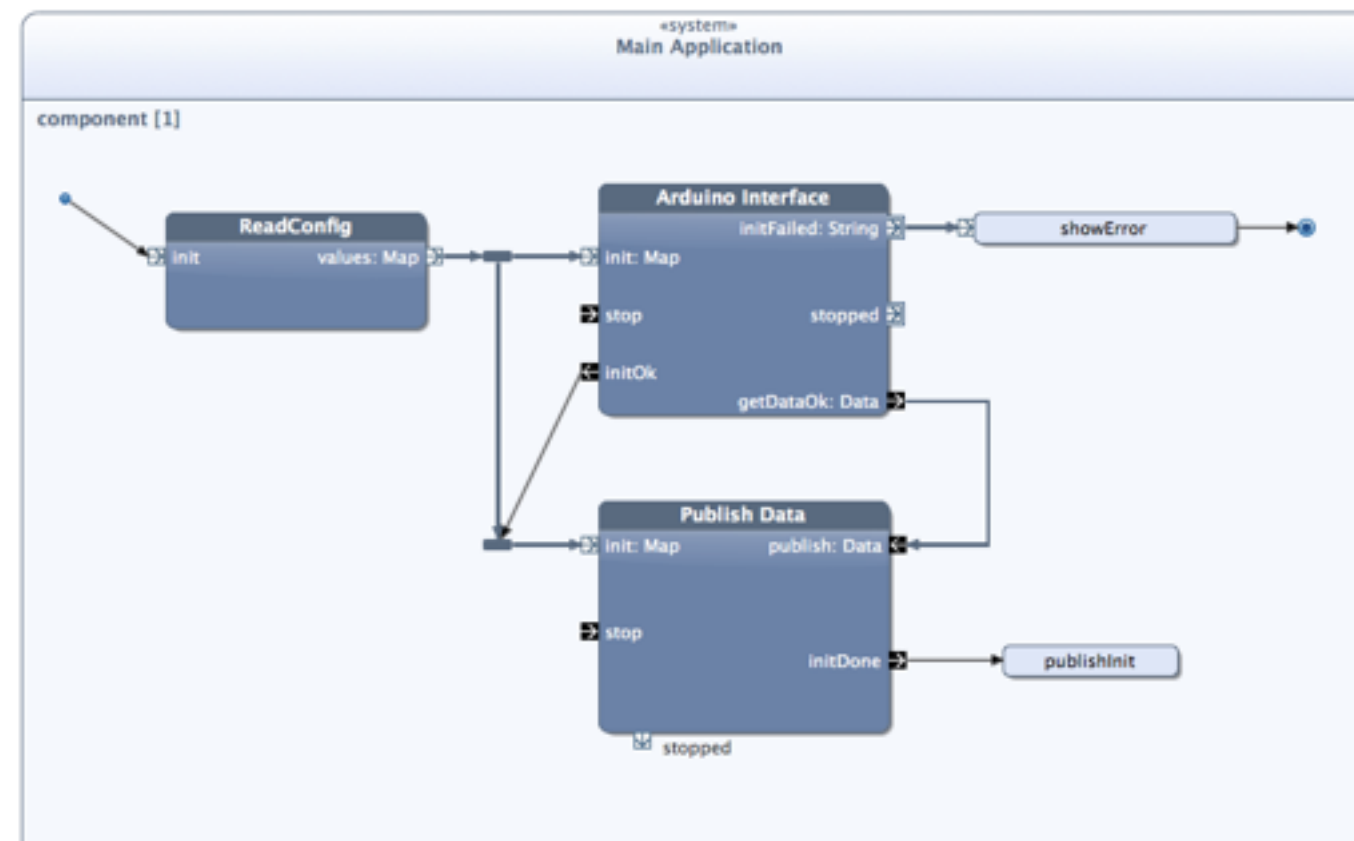
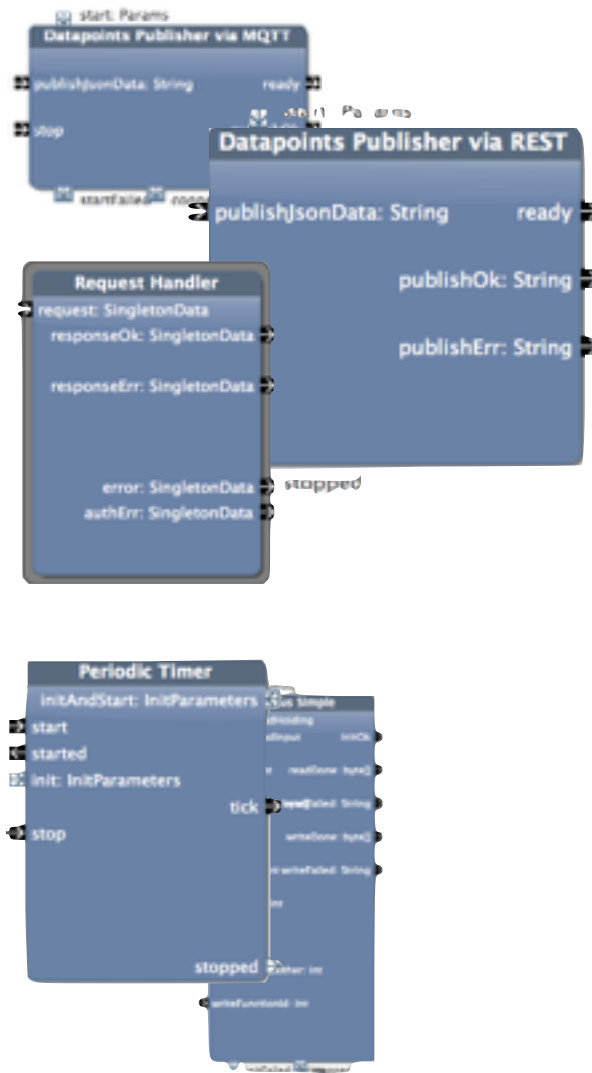


Application Development for Legato with Reactive Blocks

Frank Alexander Kraemer
frank @ bitreactive.com

Reactive Blocks

IoT / M2M Libraries



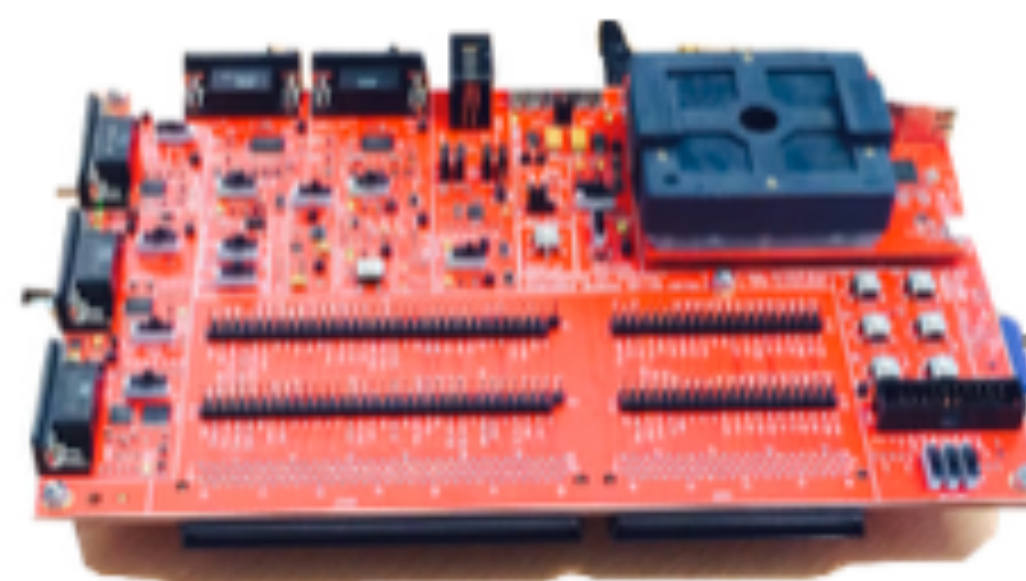
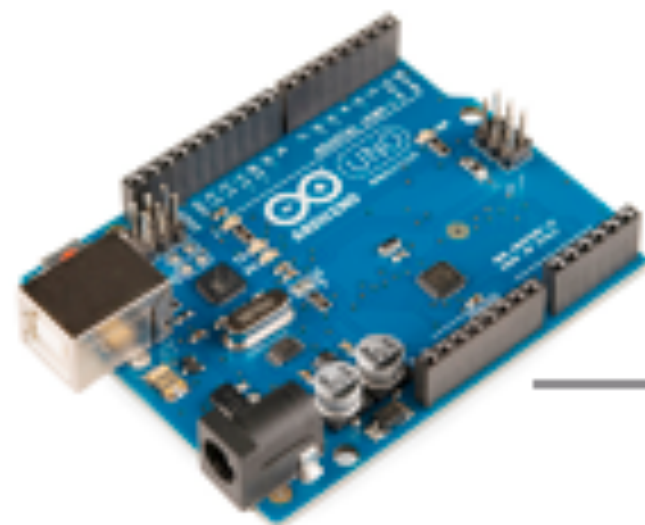
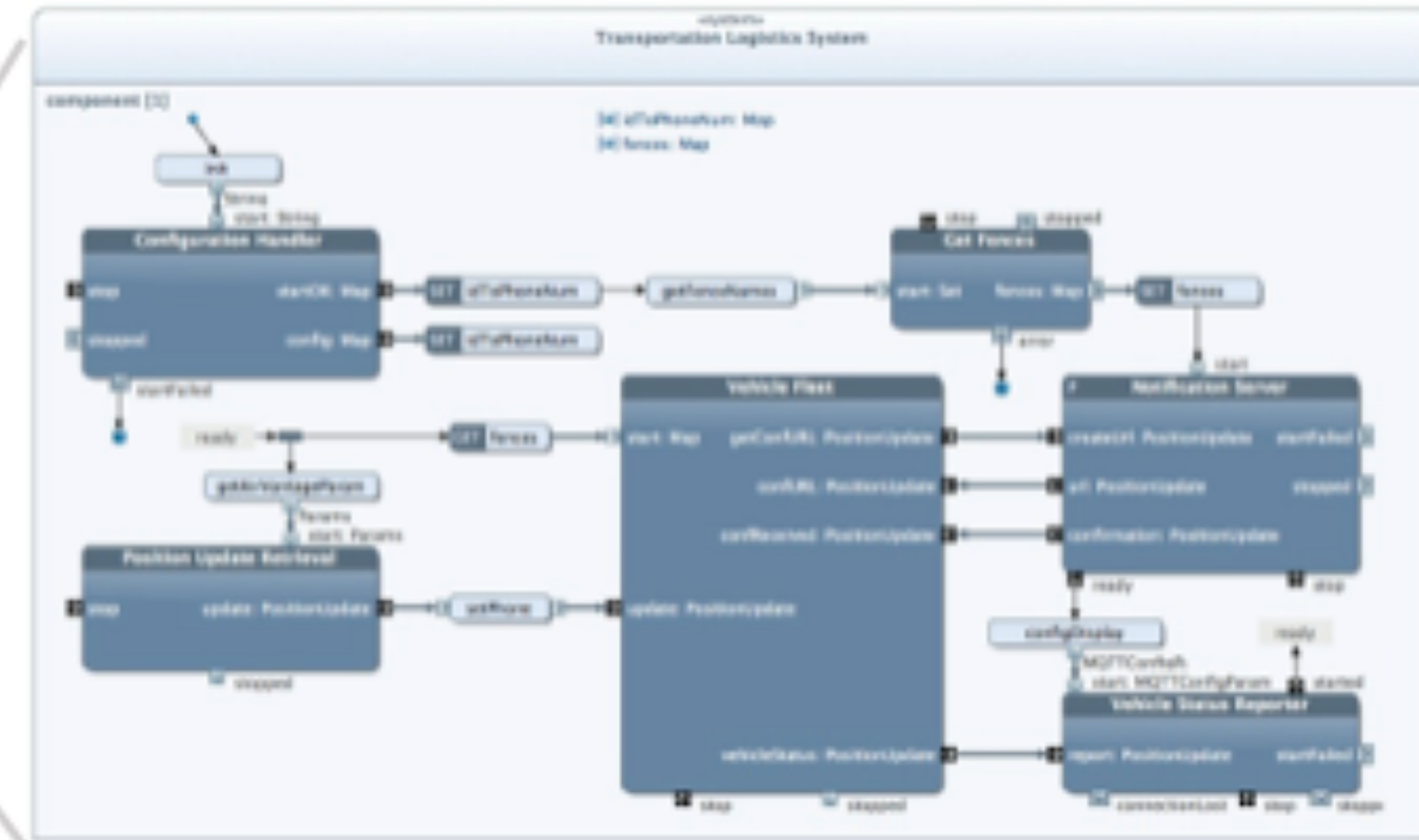
1 Pick existing blocks from the libraries

2 Build by combining blocks and Java code

3 Automatically generate code ready to deploy

Reactive Blocks

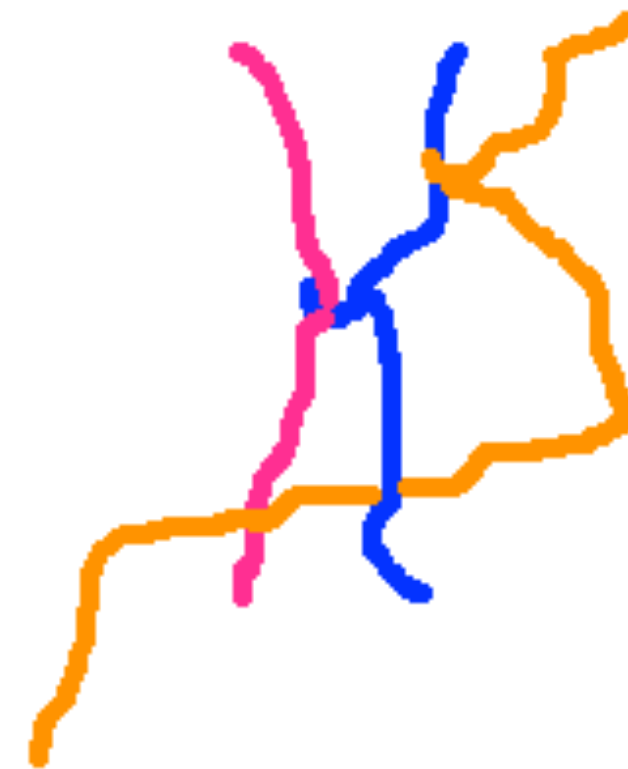
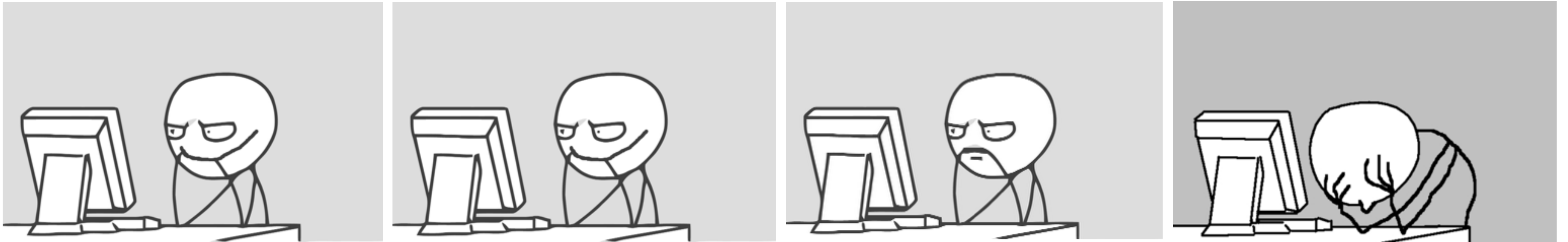
Java VM



AirVantage[®]
M2M Cloud

bitreactive

Concurrent Programming



```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.take_methere);
    goButton = (Button) findViewById(R.id.button_go);
    goButton.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText ed = (EditText) findViewById(R.id.textview);
            location = ed.getText().toString();
            Thread t = new Thread(run);
            t.start();
        }
    });
}

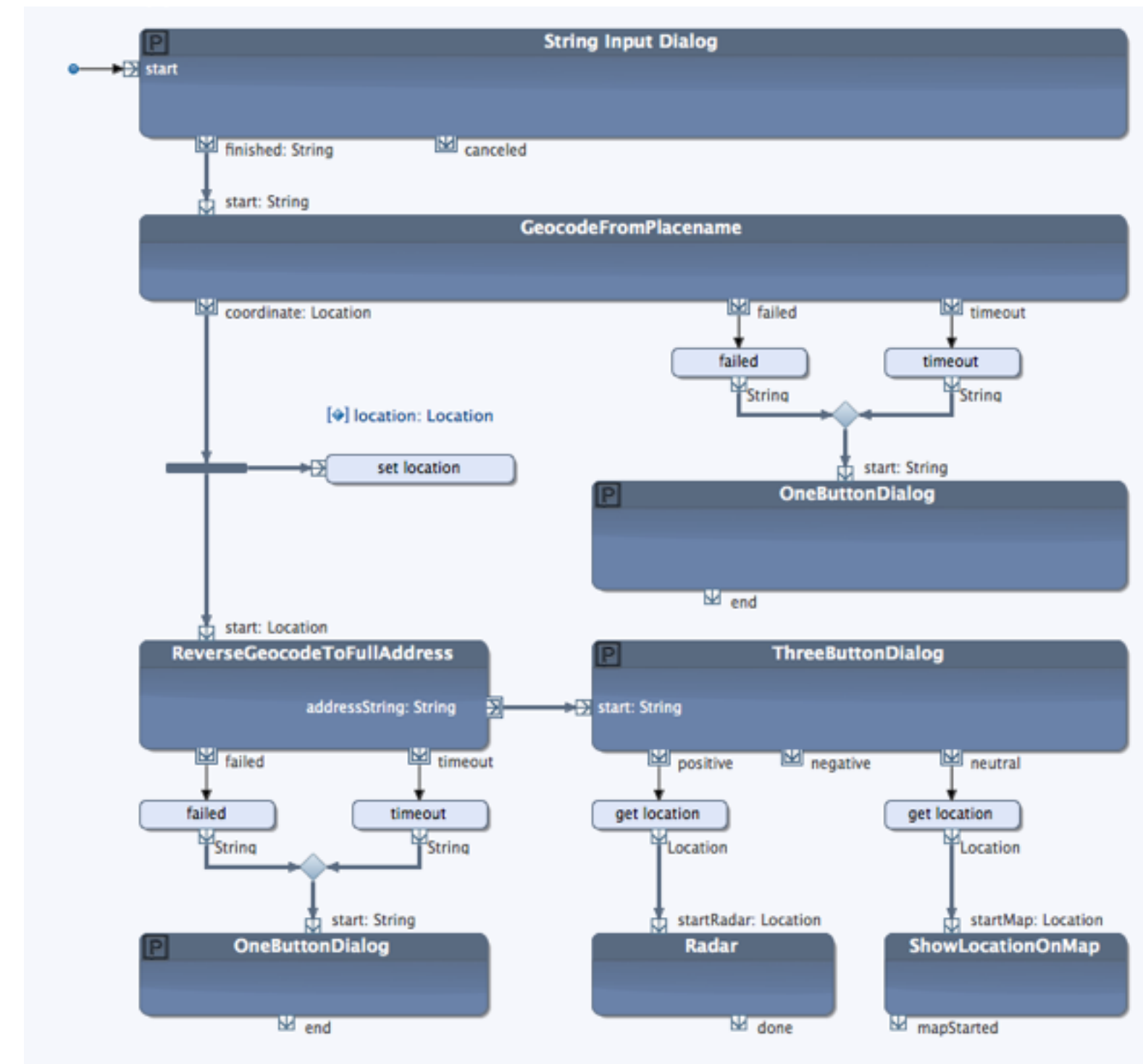
private void displayNoResultsDialog() {
    Builder builder = new AlertDialog.Builder(TakeMeThereTest.this);
    builder.setMessage("No results found");
    builder.setNegativeButton("Ok", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface arg0, int arg1) {
            builder.show();
        }
    });
    builder.show();
}

private void displayLocationDialog(String string) {
    Builder builder = new AlertDialog.Builder(TakeMeThereTest.this);
    builder.setMessage(string);
    builder.setPositiveButton("Show Radar", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface arg0, int arg1) {
            Intent intent = new Intent("com.google.android.radar.ShowRADAR");
            intent.putExtra("latitude", (float) (loc.getLatitude()));
            intent.putExtra("longitude", (float) (loc.getLongitude()));
            TakeMeThereTest.this.startActivity(intent);
        }
    });
    builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface arg0, int arg1) {
            builder.show();
        }
    });
    builder.show();
}

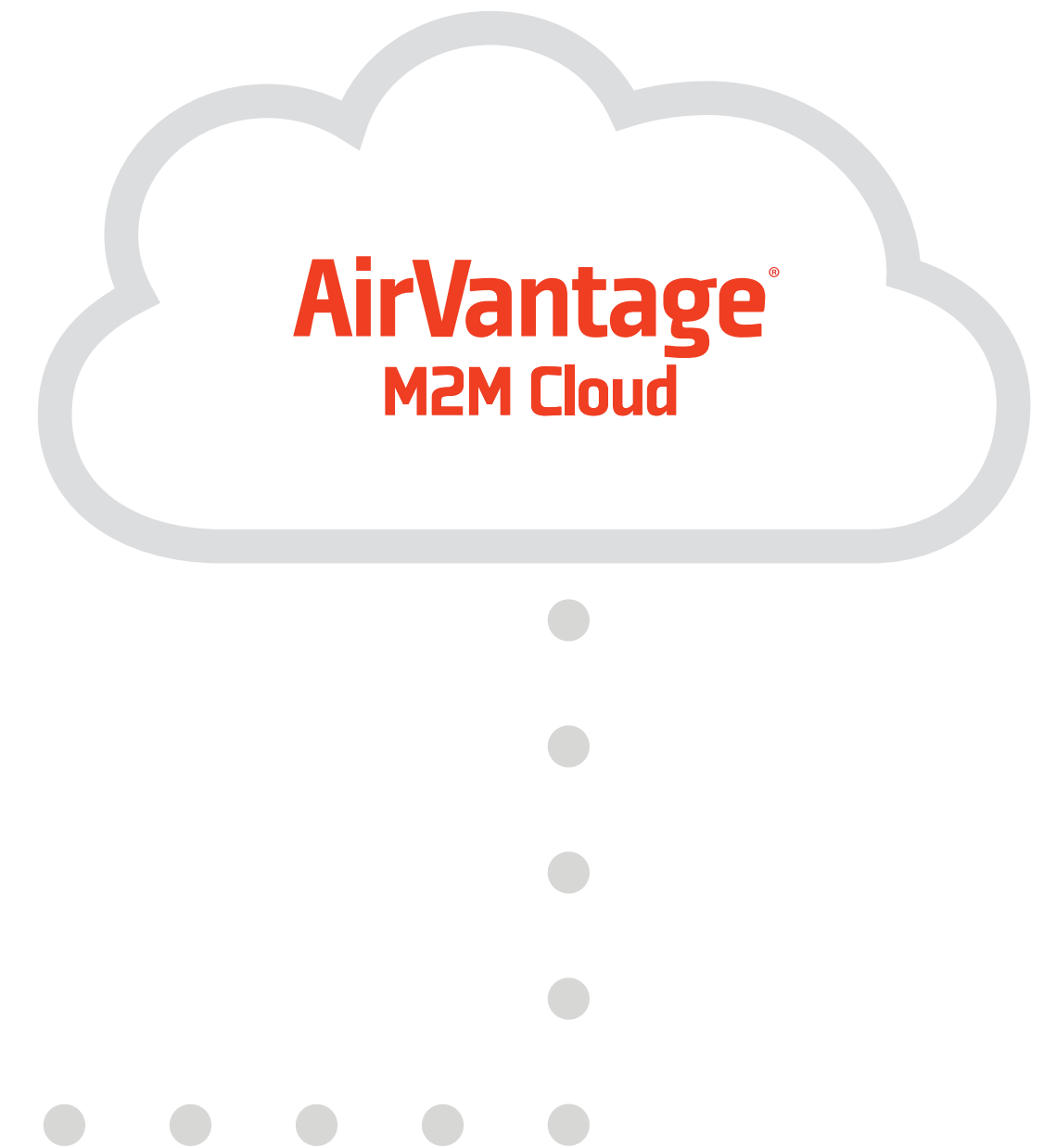
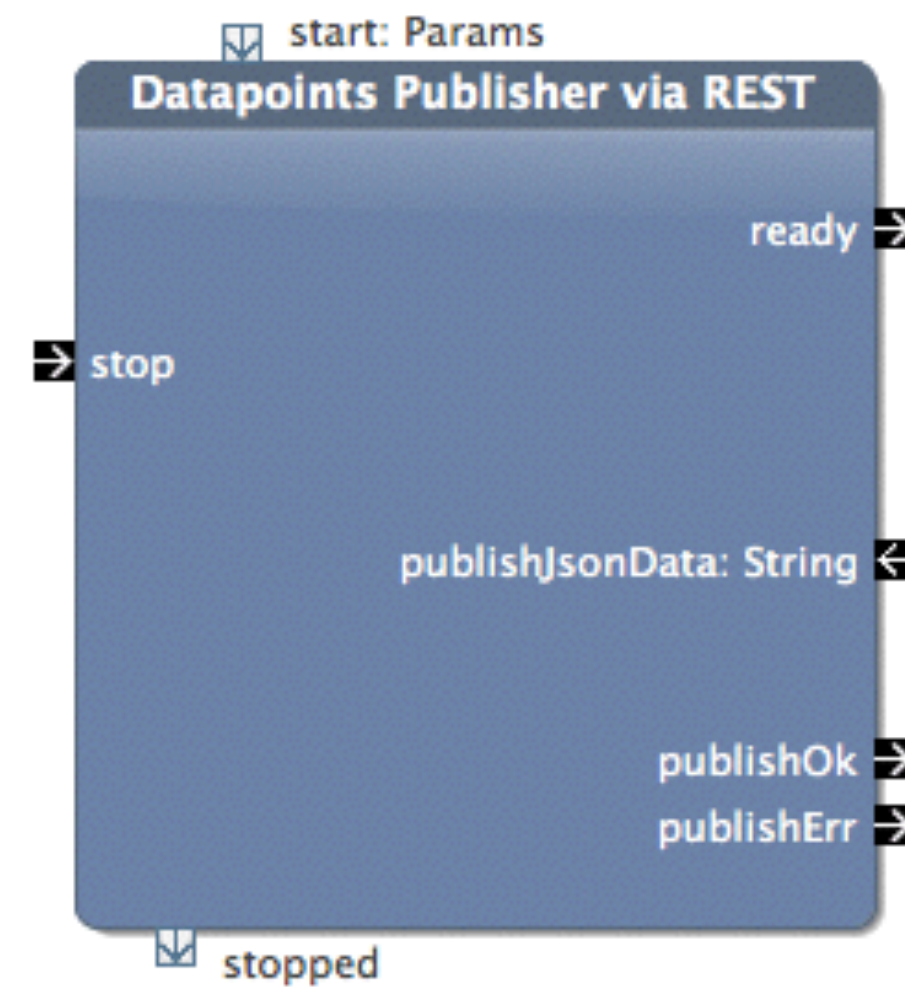
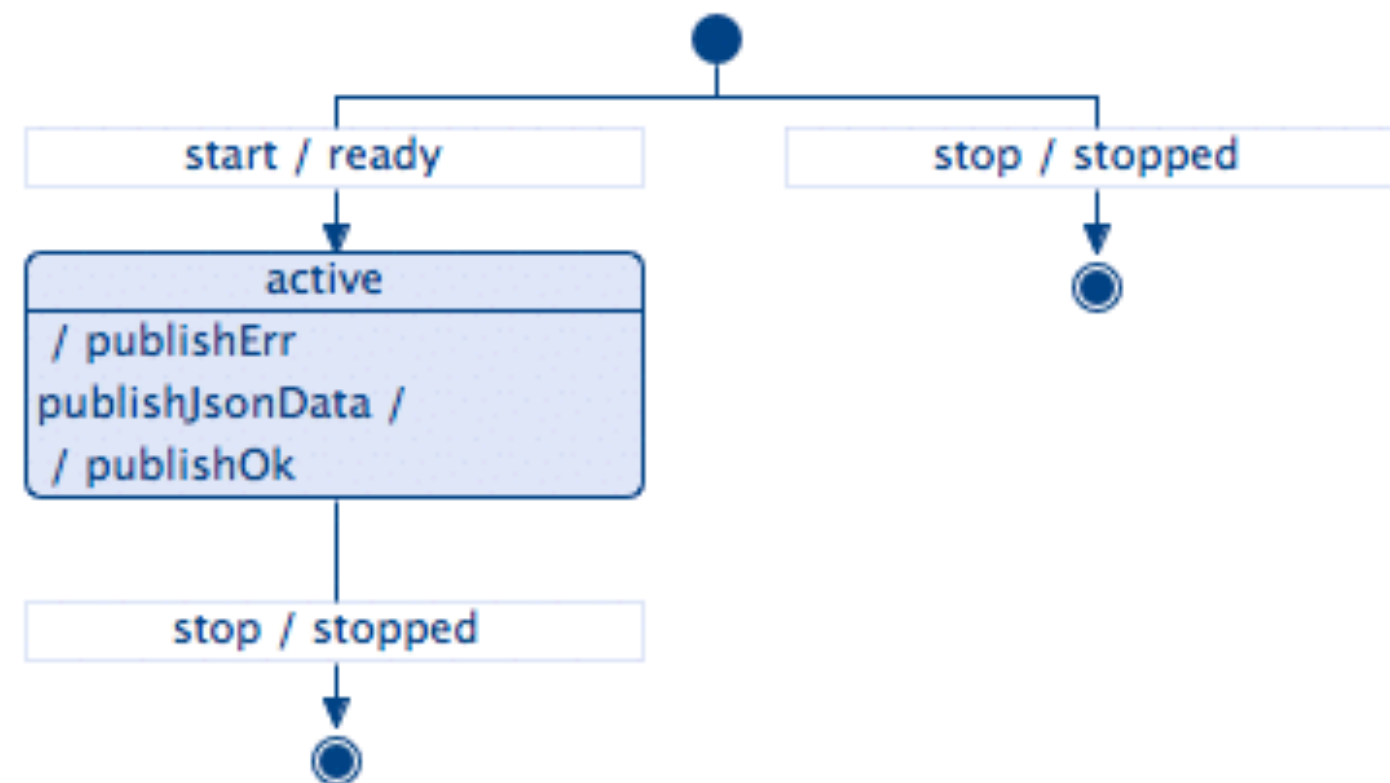
Handler updateHandler = new Handler();
public void handleMessage(Message msg) {
    switch (msg.what) {
        case 1:
            String s = (String) msg.obj;
            displayLocationDialog(s);
            break;
        case 2:
            displayNoResultsDialog();
            break;
    }
}

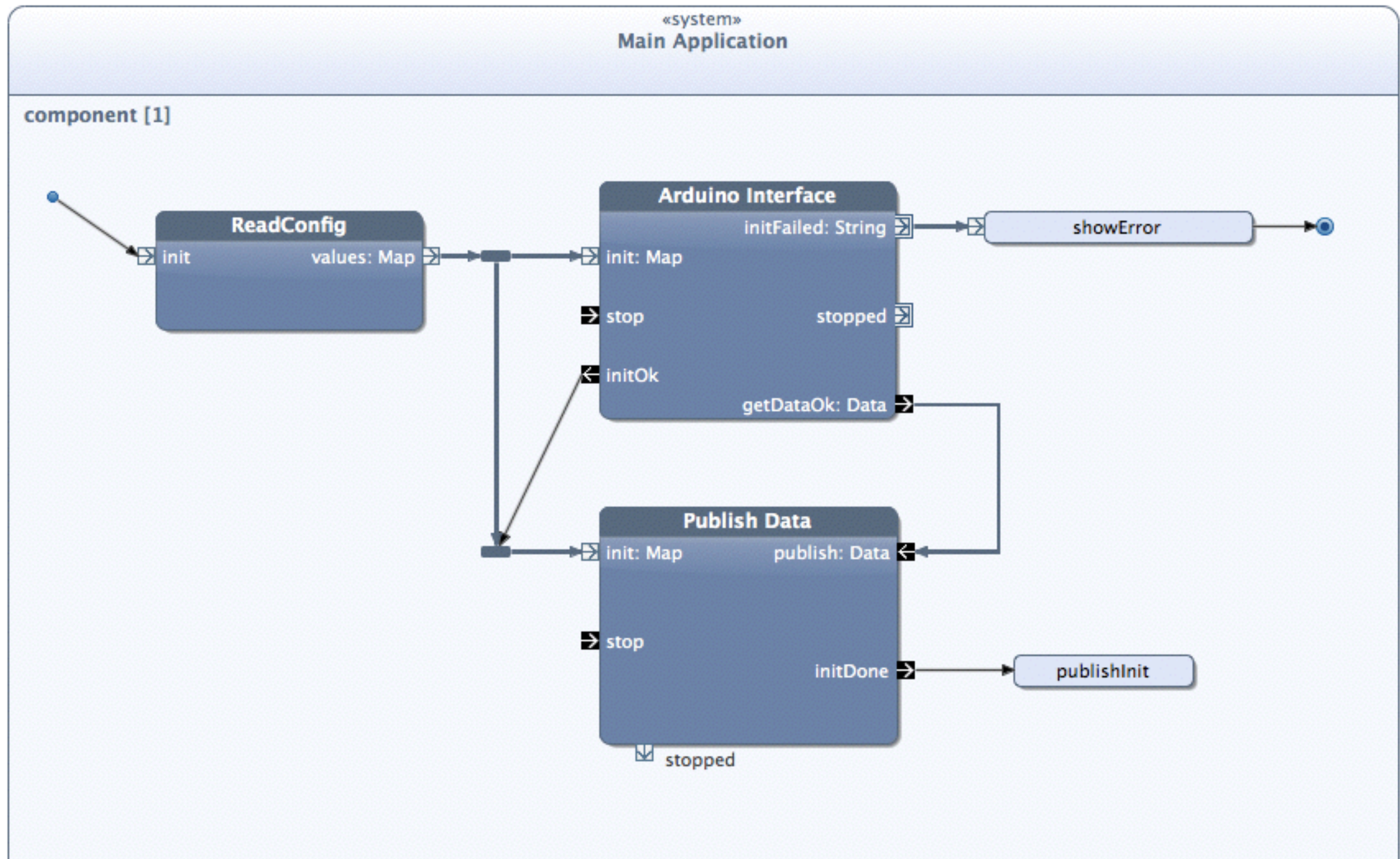
Runnable run = new Runnable() {
    public void run() {
        if (location != null) {
            Geocoder gc = new Geocoder(TakeMeThereTest.this,
                Locale.getDefault());
            try {
                List<Address> addresses =
                    gc.getFromLocationName(location, 1);
                if (addresses.size() > 0) {
                    address = addresses.get(0);
                    loc = new Location("gps");
                    loc.setLatitude(address.getLatitude());
                    loc.setLongitude(address.getLongitude());
                    List<Address> addresses2 = gc.getFromLocation(
                        loc.getLatitude(), loc.getLongitude(), 1);
                    if (addresses2.size() > 0) {
                        Address address = addresses2.get(0);
                        StringBuilder stringBuilder = new StringBuilder();
                        int lines = address.getMaxAddressLineIndex();
                        if (lines > 0) {
                            stringBuilder.append(address.getAddressLine(0) + ", ");
                        }
                        String postalCode = address.getPostalCode();
                        String cityName = address.getLocality();
                        if (postalCode != null && cityName != null) {
                            stringBuilder.append(postalCode + " " +
                                cityName);
                        }
                        Message msg = new Message();
                        msg.what = 1;
                        msg.obj = stringBuilder.toString();
                        updateHandler.sendMessage(msg);
                    } else {
                        Message msg = new Message();
                        msg.what = 2;
                        updateHandler.sendMessage(msg);
                    }
                } else {
                    Message msg = new Message();
                    msg.what = 2;
                    updateHandler.sendMessage(msg);
                }
            } catch (IOException e) {
            }
        }
    }
}

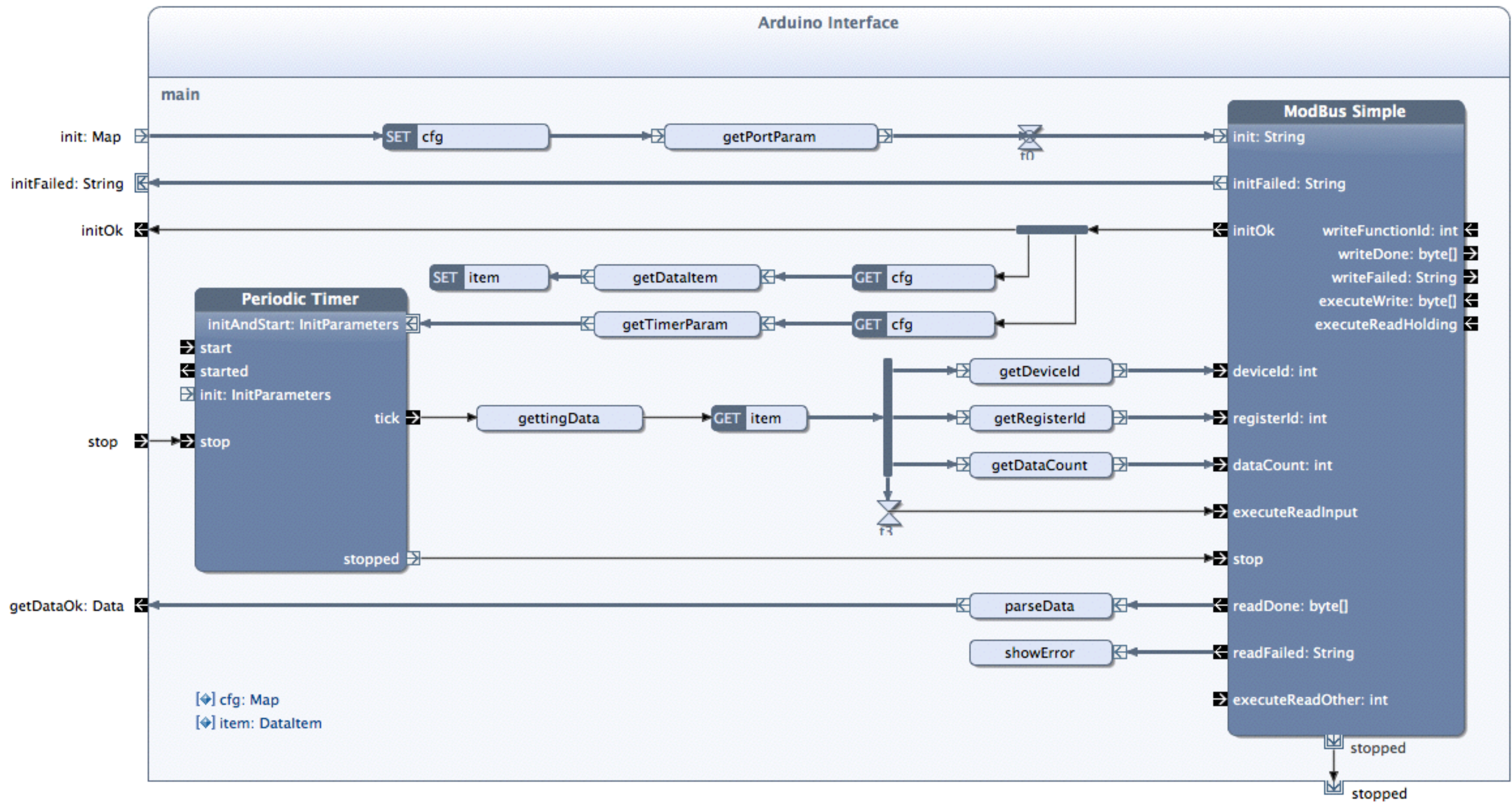
```



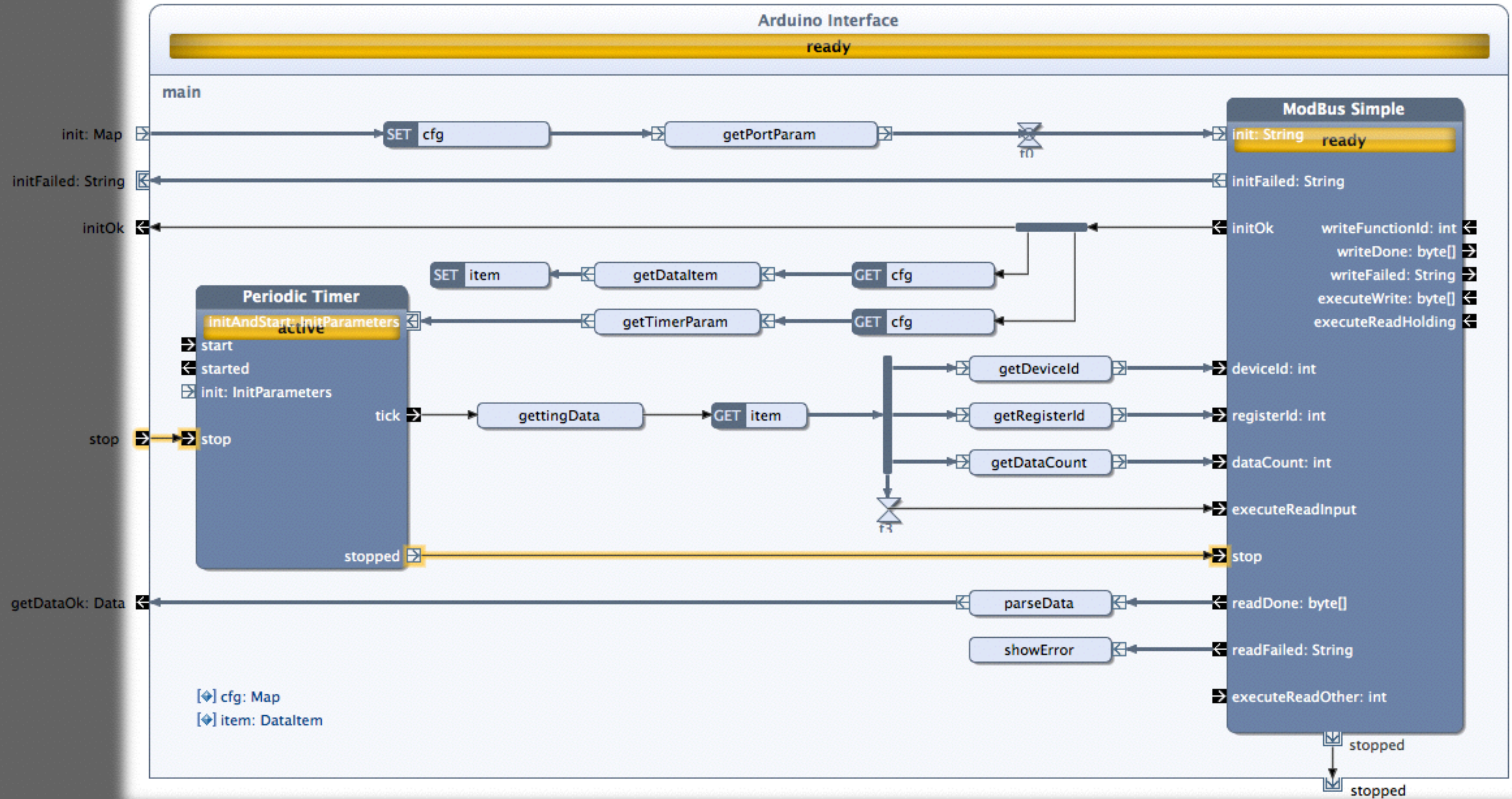
Reactive Blocks

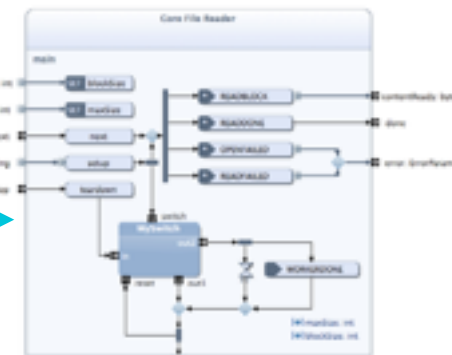
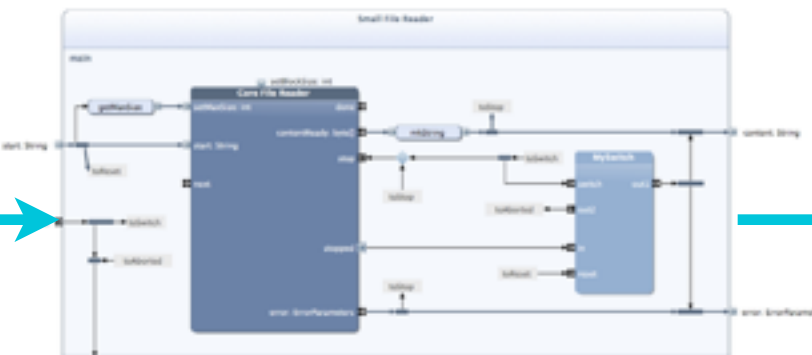
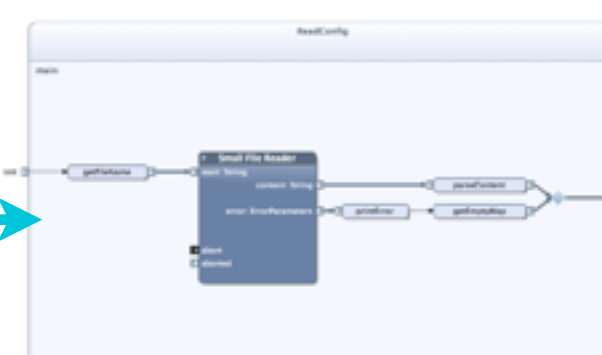
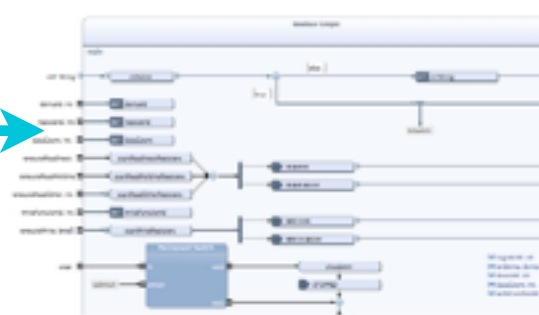
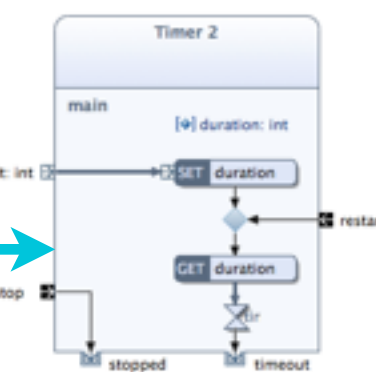
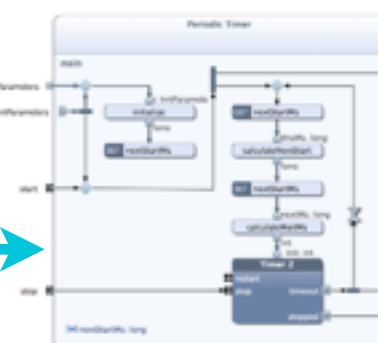
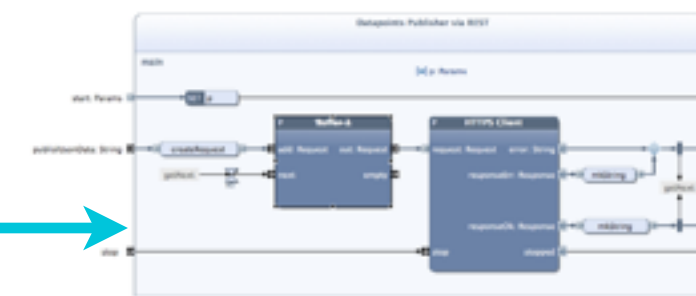
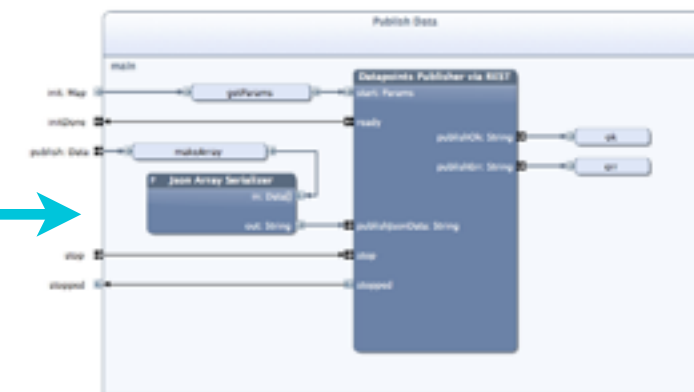
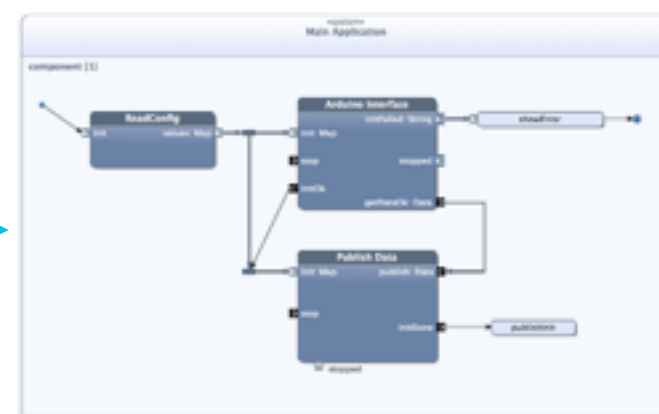
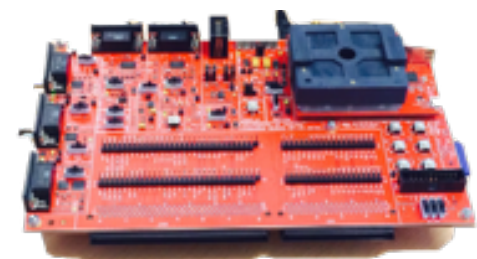




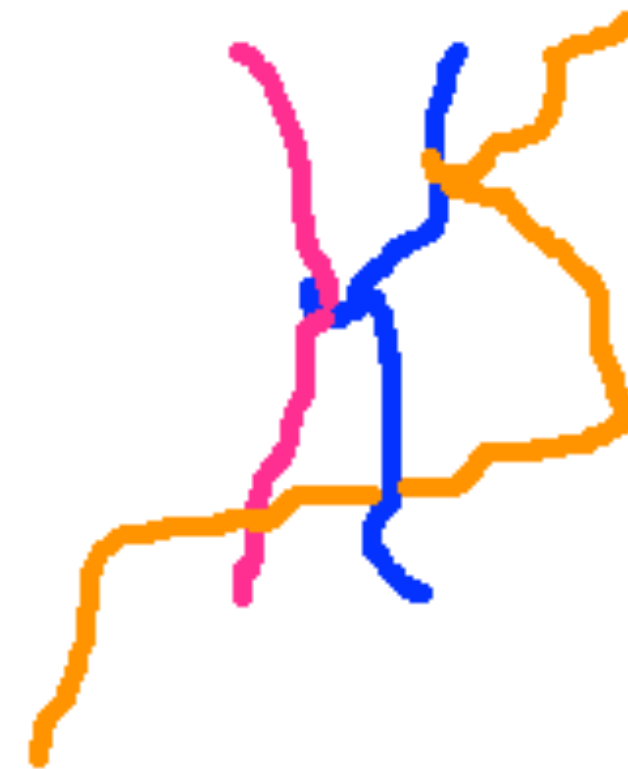
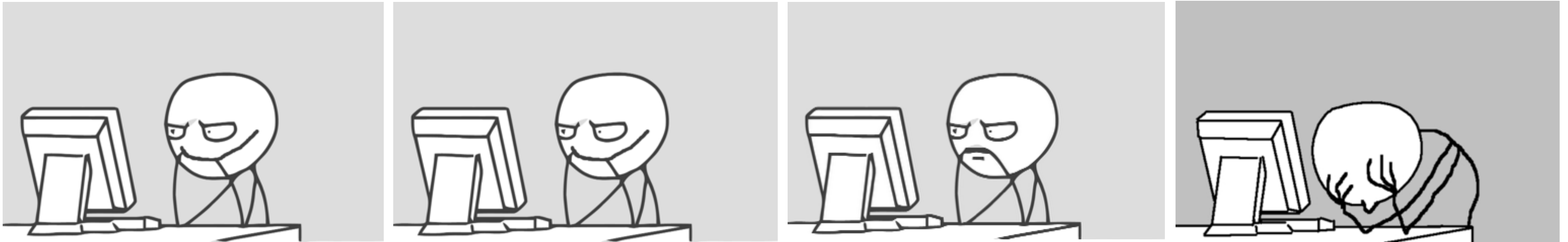


'x' to exit



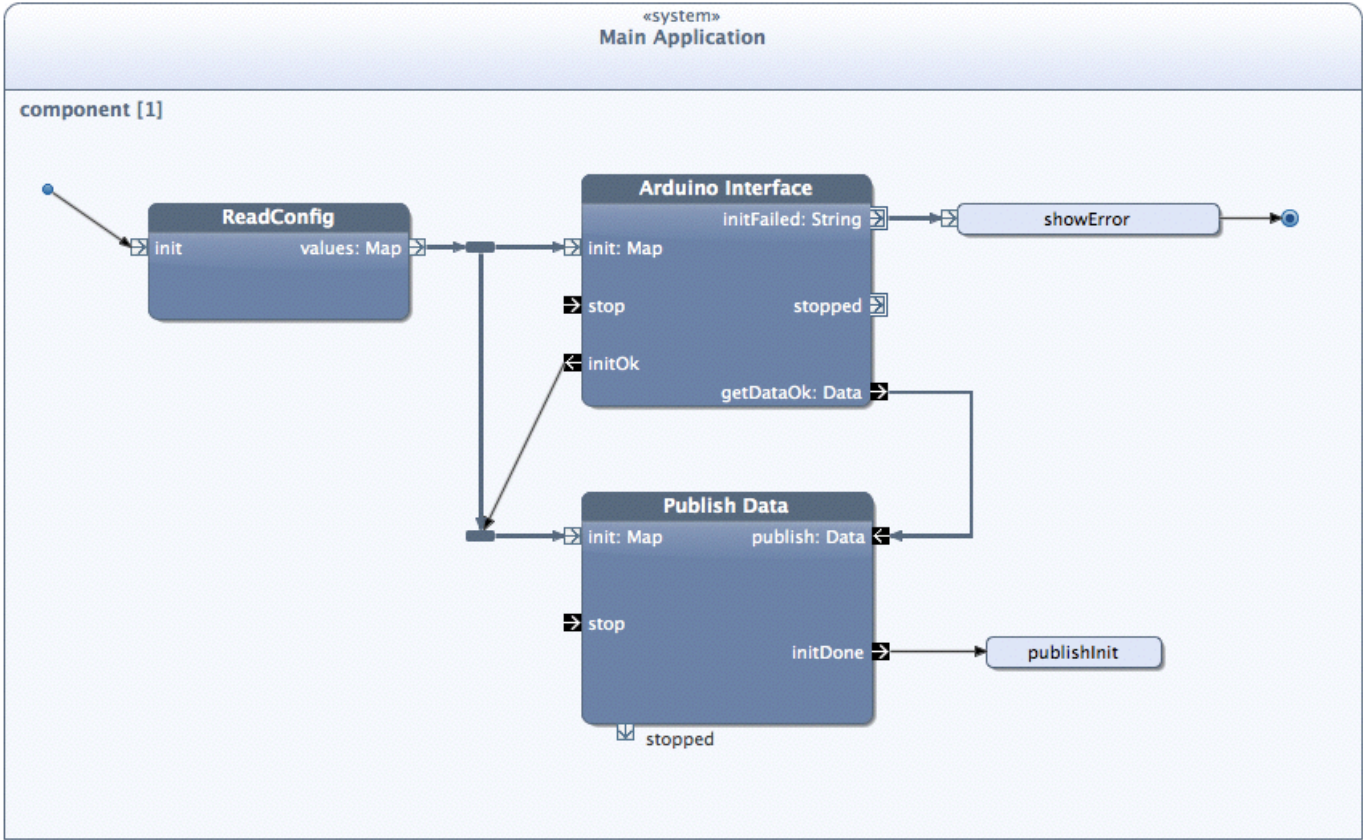
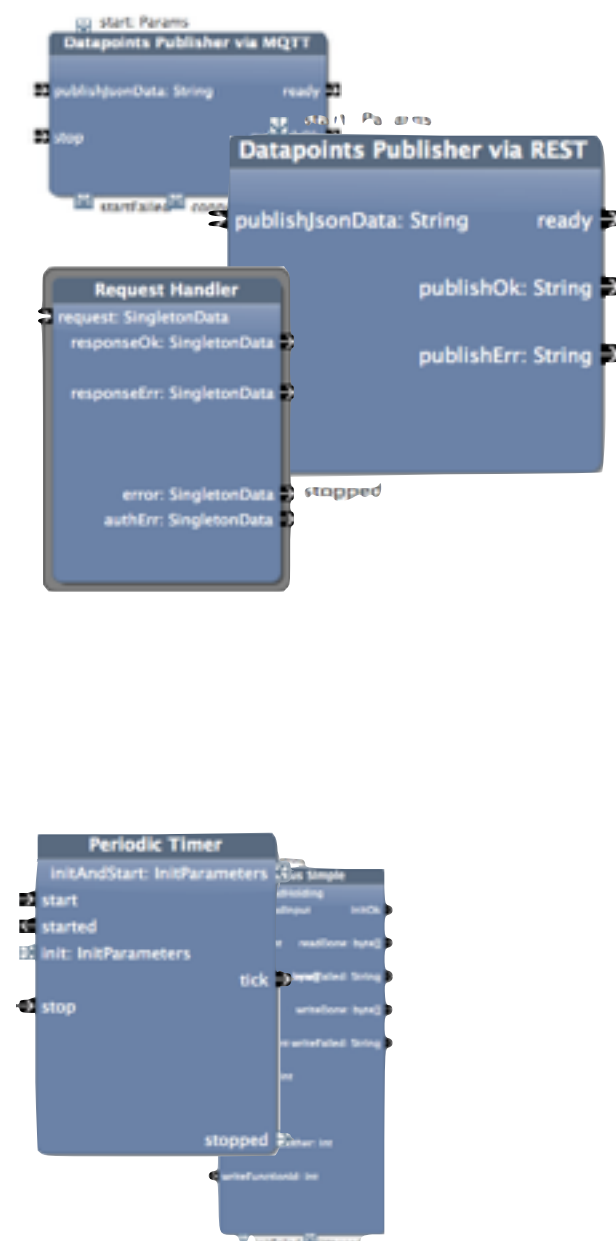


Concurrent Programming



Concurrent Programming

IoT / M2M Libraries



- 1

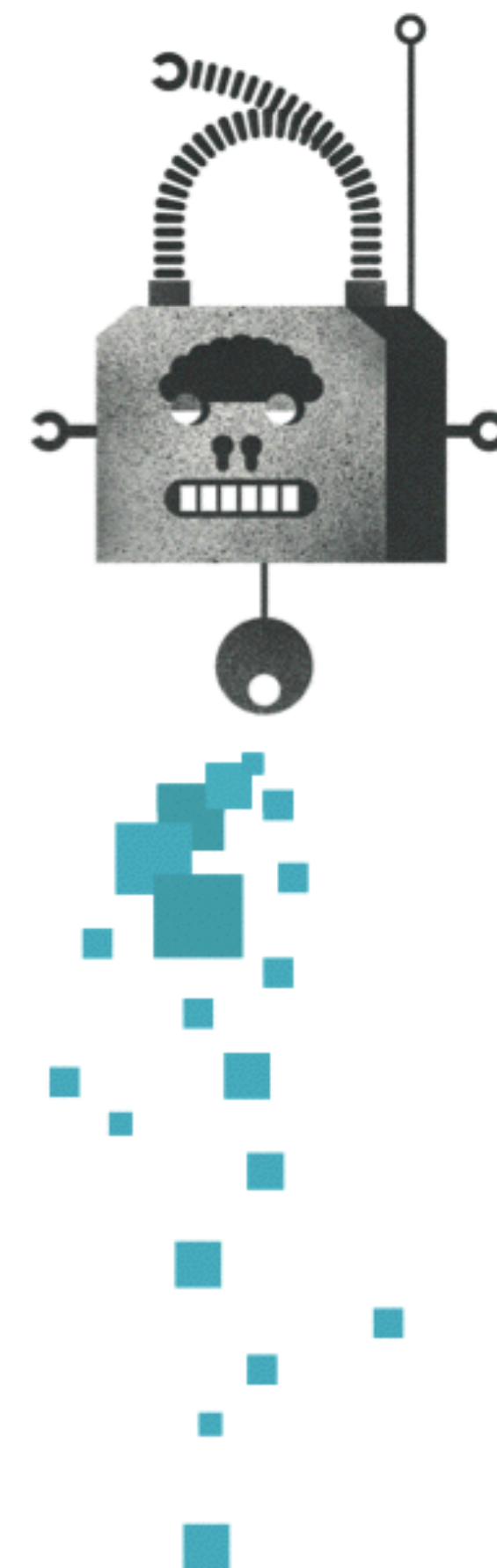
Pick existing blocks from the libraries
- 2

Build by combining blocks and Java code
- 3

Automatically generate code ready to deploy



bitreactive.com



bitreactive